UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Estimación y caracterización de la demanda en un bike-sharing system universitario utilizando Machine learning

TESIS

Que para obtener el título de

Ingeniero Industrial

P R E S E N T A

Bruno Varela Petrissans

DIRECTORA DE TESIS

Dra. Esther Segura Pérez

Ciudad Universitaria, Cd. Mx., 2021

A Luz, Edgard y Bárbara

Con mucho cariño y dedicación

Agradecimientos

A la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería quiénes me brindaron los recursos posibles para mi desarrollo profesional y la oportunidad de conocer a docentes comprometidos con la educación y el desarrollo de la sociedad.

Al sistema de transporte Bicipuma, el cuál me ofreció todo su apoyo para la realización de esta investigación, de lo contrario, este proyecto no hubiera sido posible.

A mi familia: Luz, Edgard y Bárbara, quiénes me han apoyado en todo momento en mis proyectos y objetivos. Quiénes siempre me han inspirado a trabajar y ser una mejor persona cada día. Gracias.

A mi directora de tesis, Dra. Esther Segura, ya que su profesionalismo, dedicación y compromiso con la investigación científica, determinó e hizo posible el desarrollo de esta investigación.

Muchas gracias.

Resumen

Como un transporte sustentable y alterno a los medios convencionales, los sistemas de bicicletas compartidas (*Bike-sharing systems* o *BSS*, por sus siglas en inglés) han aumentado su popularidad debido a sus beneficios como cero emisiones, viajes incluso más rápidos que los transportes convencionales, fomentar el ejercicio, además, de ayudar a mitigar el tráfico en las calles, la contaminación del aire y el ruido.

Al igual que en las ciudades, las universidades han implementado *BSS* para ayudar a la comunidad universitaria a transportarse por todo el campus. Bicipuma fue el primer *BSS* en la Ciudad de México y el primer *BSS* universitario en México. Diseñado para el transporte y el uso recreativo, Bicipuma se ha convertido en uno de los medios de transporte más populares de la UNAM con 14 estaciones de bicicletas, 980 bicicletas y alrededor de 4,000 viajes diarios.

Un problema común detectado en muchos sistemas de bicicletas compartidas, así como en Bicipuma, es la escasez o los excedentes que se presentan en las estaciones de bicicletas, por lo que, se reequilibran manualmente con camionetas, aumentando los costos operativos del sistema y, en muchas ocasiones, no satisfaciendo la demanda de los usuarios.

Esta investigación describe y propone un modelo para predecir la demanda de Bicipuma con dos algoritmos de *Machine Learning*, Random Forests y XGBoost, analizando más de dos millones de observaciones, los cuales consisten en datos registrados por Bicipuma desde enero de 2017 hasta noviembre de 2019.

Este trabajo se desarrolla siguiendo una metodología de ciencia de datos de 10 pasos para explicar todo el proceso de predicción y caracterización de la demanda de *BSS*, analizando datos históricos y haciendo una selección de variables para el modelo de *Machine Learning*.

Los resultados obtenidos de esta investigación indican que, dividir la variable de tiempo en minutos hasta años, y considerar algunas variables climáticas, son muy importantes para un mejor modelado del sistema. Se aplicaron los algoritmos Random Forests y XGBoost para predecir la demanda de cada enlace del sistema, el error cuadrático medio (RMSE, por sus siglas en inglés)

para los movimientos diarios de la bicicleta en cada enlace entre estaciones, está en un rango de uno a ocho bicicletas, una predicción que es aceptable para ser utilizado por el sistema Bicipuma.

Abstract

As an alternative form of transportation, *bike-sharing systems* (*BSS*) have increased their popularity due to their benefits like zero emissions, healthy and even faster commuting than conventional transportation, moreover, helps to mitigate road congestion, air, and noise pollution. As well as in the cities, universities have implemented *BSS* to help the university community to transport throughout the campus. As a pioneer, Bicipuma was the first *BSS* in Mexico City and the first university *BSS* in Mexico. Designed for transportation and recreational usage, Bicipuma has become one of the most popular transportation methods in UNAM with 14 bike stations, 980 bicycles, and around 4,000 trips a day.

A common problem detected in many *bike-sharing systems*, as well as in Bicipuma, is the shortage or the surpluses presented in bike stations, which are manually rebalanced with trucks, increasing operational costs of the system, and not meeting the demand of the users.

This investigation describes and proposes a model to predict the demand of Bicipuma with two *Machine Learning* algorithms, Random Forests, and XGBoost, analyzing more than 2 million observations, consisting of data registered by Bicipuma from January 2017 to November 2019.

Throughout this thesis, a 10-step data science methodology is followed to explain the entire process to predict and characterize the *BSS* demand, analyzing historical data, and making a variable selection for the *Machine Learning* model, a crucial activity to understand the behavior of the system and, as a first step to make a structure replenishment plan.

The results from this study indicate that splitting the time variable into minutes until years, and consider some weather variables, are very important for better modeling of the system. Random Forests and XGBoost were applied to predict the demand of every link of the system. The average RMSE for daily bike movements in every link is in a range of 1 to 8 bikes, a useful practical prediction.

CONTENIDO

Capítulo 1

Introducción

Historia de los sistemas bike-sharing

Los sistemas *bike-sharing* han incrementado su popularidad mundialmente como una alternativa de transporte para reducir problemas comunes de una ciudad urbanizada como contaminación auditiva y del aire, así como mejorar los traslados, reduciendo la congestión vial. (Ashqar et al., 2019). Cabe destacar que los sistemas *bike-sharing* comenzaron a popularizarse en la década de los 2000, donde este modo de transporte creció de 13 programas en 2004 hasta 855 alrededor del mundo en 2014. Uno de los programas más grandes como lo es el Vélib de Paris, ha registrado hasta 86,000 viajes por día (Embarq network, 2021).

Sin embargo, la historia de este sistema se remonta a la década de 1960 cuando se estableció en Ámsterdam el primer sistema *bike-sharing*, llamado "Witte Fietsen", que consistía en la colocación de 2000 bicicletas distribuidas por toda la ciudad. El usuario podía tomar una bicicleta para ir de un punto A a un punto B, dejándola libre para el siguiente usuario. El sistema de primera generación resultó ser muy poco eficiente debido a la dificultad para encontrar bicicletas disponibles y a la susceptibilidad de estas, a ser robadas (Datta, 2014).

La segunda generación de los sistemas *bike-sharing* se conceptualizó en Dinamarca en 1991, pero no fue sino hasta 1995 que se creó el primer sistema *bike-sharing* a gran escala en la capital de dicho país. Subsecuentemente, hubo numerosas mejoras en comparación a la primera generación de estos sistemas, tales como bicicletas con diseños más resistentes y locaciones específicas en donde el usuario hacía un depósito que era reembolsado al final de su viaje a través de un básico sistema de depósito de monedas. No obstante, en este sistema aún era fácil que las bicicletas fueran robadas, en parte importante debido a que no se tenía un sistema de seguimiento de los viajes realizados por los usuarios (DeMaio, 2009).

Para 1996 se estableció en la Universidad de Porstmouth el primer sistema *bike-sharing* de tercera generación. Esta generación se caracteriza por identificar a los usuarios y utilizar tecnología en

todo el proceso tal como puertos electrónicos, sistemas de comunicación, acceso al servicio vía smartphone, y pago de comisión con tarjeta de crédito, entre otras (DeMaio, 2009). La tercera generación ha crecido significativamente desde su creación. Esta alternativa de transporte ha sido adoptada con gran aceptación en numerosas ciudades alrededor del mundo. En muchos casos, el crecimiento de cada sistema crece de manera orgánica, es decir, crece de acuerdo con los requerimientos del servicio. (O'Brien, 2014).

Los sistemas *bike-sharing* suelen funcionar como transporte de primera o última milla (First mile / Last mile), es decir, las personas viajan en bicicleta para tomar un transporte público de larga distancia o, para llegar a su destino después de tomar un transporte de larga distancia (Shaheen et al., 2014).

Cada ciudad adopta un sistema *bike-sharing* acorde a los requerimientos de movilidad, su urbanización o plan de desarrollo urbano, y al presupuesto destinado al crecimiento de transportes sustentables. Por lo tanto, han surgido distintos modelos de sistemas *bike-sharing* en función de (DeMaio, 2009):

- Gobierno
- Agencia de transporte (cuasi-gubernamental)
- Organizaciones sin fines de lucro
- Compañía publicitaria
- Organización con fines de lucro
- Universidad

Como ejemplo del último modelo mencionado está el sistema *bike-sharing* de la Universidad Nacional Autónoma de México, llamado Bicipuma.

El sistema Bicipuma fue creado en 2004 cuando la Facultad de Medicina de la UNAM decidió establecer una estación de bicicletas cerca de sus instalaciones y otra estación cerca del estadio Olímpico Universitario. La distancia entre estas dos estaciones era corta y su uso solo con fines recreativos. Un año después, se estableció un sistema *bike-sharing* con siete estaciones alrededor

del campus Universitario (UNAM, 2018), siendo este el primer sistema de su clase en la Ciudad de México y el primero en una universidad mexicana (Travesía UNAM, 2020).

El principal motivo para crear este sistema *bike-sharing* fue la gran afluencia de estudiantes en el campus universitario. Basado en datos de marzo de 2019, se habían registrado trescientos mil visitantes por día entre estudiantes, profesores, empleados de la universidad, investigadores y público en general (UNAM, 2019). Dicha afluencia causa embotellamientos e impacta directamente a los sistemas de transporte universitarios entre ellos Bicipuma y Pumabus, consecuentemente, estos transportes mitigan el problema y ayudan a la comunidad universitaria a transportarse de forma fácil a través del campus.

Con datos registrados hasta 2019, Bicipuma cuenta con 14 estaciones de bicicletas, 980 bicicletas en funcionamiento, ocho kilómetros de ciclovía y alrededor de 4,000 viajes diarios. Es uno de los medios de transporte gratuitos que se ofrece a la comunidad universitaria, promoviendo la actividad física y la sostenibilidad ambiental (UNAM, 2017).

Lo que se requiere para ser un usuario de Bicipuma es, tener una credencial UNAM vigente y una contraseña de cuatro dígitos. Estos dos requisitos ayudan a mejorar la trazabilidad de cada viaje a través de un sistema básico de lectura de código de barras.

Descripción del problema

Como uno de los principales medios de transporte dentro del campus universitario de la UNAM, Bicipuma ha registrado alrededor de 4,000 viajes al día durante el año 2019, por lo que es común ver desabastecimiento y excedente de bicicletas en cada estación de bicicletas del sistema durante el horario de operación de Bicipuma. Por ello, el personal de Bicipuma debe reabastecer o retirar bicicletas en algunas estaciones actuando como agentes externos al flujo natural de bicicletas, es decir, el personal de Bicipuma utiliza camionetas para intentar equilibrar el nivel de bicicletas de las 14 estaciones y satisfacer la demanda de los usuarios.

Una empresa, industria, o proyecto, pueden ser considerados como un sistema; cuando un sistema es ineficiente siempre hay una manera de mejorar su funcionamiento, es decir, el sistema podría

optimizarse. Sin embargo, cuando un sistema funciona sujeto a un presupuesto, el costo de la ineficiencia podría ser crítico y podría causar el cierre de un proyecto o no alcanzar el objetivo para el cuál se creó (Sickles y Zelenyuk, 2019). Esa es la razón por la que optimizar los recursos y satisfacer la demanda de un sistema es tan importante para Bicipuma y otros sistemas *bike-sharing* que se rigen como una organización sin fines de lucro y que están sujetas a un presupuesto.

Según Ernesto García Almaraz, coordinador del sistema Bicipuma, el proceso de reabastecimiento actual se realiza de forma manual y, los patrones de demanda de cada estación se han aprendido de manera empírica, por lo que el personal del programa no tiene un conocimiento estructurado para realizar un plan de reabastecimiento para cada estación de bicicletas de acuerdo al tiempo de operación del sistema, lo que provoca una respuesta ineficaz a la demanda de los usuarios o reabastecimientos innecesarios de bicicletas entre las estaciones con baja demanda, lo que genera un aumento de los costos operativos.

Como primer paso para mejorar la eficiencia de un *BSS* (*Bike-sharing system*), es fundamental entender las variables que afectan al sistema y descubrir los patrones de demanda, comprender el sistema de manera analítica, para luego poder predecir la demanda de la red Bicipuma.

Objetivo general

Caracterizar y predecir la demanda de un sistema de *Bike-sharing* universitario a través de una metodología de ciencia de datos, utilizando herramientas de Machine learning.

Objetivos específicos

1. Identificar el impacto de las variables climatológicas en la demanda del sistema
2. Descubrir los patrones de demanda del sistema
3. Comparar el desempeño de algunas herramientas de Machine learning, en áreas de tiempo de entrenamiento del modelo, error de pronóstico y tiempo de respuesta.

Justificación de la investigación

Esta investigación se realiza debido a la necesidad del análisis de la gran cantidad de datos generados en un sistema *Bike-sharing* universitario, y los cuales no son analizados ni utilizados para un fin benéfico hacia el sistema, es decir, utilizar los datos generados para una mejor toma de decisiones en el sistema, tal como esta investigación plantea el primer paso sobre el uso de datos para la optimización de un sistema.

En la actualidad, los datos están en todas partes, cada proceso digitaliza mediciones, observaciones, acciones, conteos, etcétera, y el nivel de acumulación de estos datos es tan grande que es un recurso importante para mejorar y optimizar procesos dentro de un sistema. Es por esta razón que esta investigación se centra en el análisis de los datos generados en un sistema *Bike-sharing* para la caracterización de su demanda, el cuál es el punto de partida para una transformación digital en el ámbito de transporte, es decir, una transformación hacia un sistema de transporte inteligente (ITS, por sus siglas en inglés).

Investigaciones relacionadas

En los últimos años, los sistemas *bike-sharing* tienen una tendencia creciente de popularidad, por consecuencia, se han generado muchas investigaciones y estudios sobre el tema (Moncayo-Martínez, 2020), muchos de ellos centrados en el análisis de los datos que proporciona un *BSS* y como principales objetivos, el mejorar el servicio del sistema, procurar satisfacer la demanda tratando de evitar desabastecimientos o excedentes, para conseguir que más personas utilicen este medio de transporte (Olvera et al., 2018).

Muchos modelos e investigaciones se basan en predecir movimientos en el sistema, número de estaciones individuales de bicicletas disponibles, número de bicicletas en cada estación, entre otras variables, con el mismo objetivo, satisfacer la demanda del sistema *bike-sharing*. De acuerdo con las investigaciones, los principales objetivos de estos modelos han sido obtener nuevos conocimientos y correlaciones entre la demanda de bicicletas y otros factores, como fecha y hora de operación, variables climatológicas, entre otros; para comprender qué variables impactan en el sistema y, ayudar al personal de estos sistemas a tomar mejores decisiones. (Ashqar et al., 2019).

Referente a la predicción de los modelos propuestos, análisis de demanda y pronóstico de los datos obtenidos en estos tipos de sistema, algunas investigaciones (Ashqar et al., 2019; Wang and Kim, 2018; y Schuijbroek et al., 2017), toman el número de estacionamientos de bicicletas individuales vacíos como una variable objetivo, algunos otros (Datta, 2014) consideran el número de bicicletas por cada estación como resultado del modelo.

En contraste con los modelos mencionados, este documento considera el número de viajes realizados entre las estaciones como variable objetivo, lo cual será explicado posteriormente en la sección Metodología – variable dependiente N análisis de la respuesta del modelo.

(Ashqar et al., 2017) propuso un modelo para predecir el número de bicicletas disponibles en cada estación a través del tiempo, en el cual se utilizaron los algoritmos *Random Forests* y *Least-Squares boosting*. Las variables más importantes consideradas para este modelo fueron: identificación de la estación, número de bicicletas disponibles, mes, día de la semana, número de bicicletas disponibles en estación *i* en el tiempo *t*, número de bicicletas disponibles en estaciones vecinas en el tiempo *t*, temperatura promedio, humedad promedio, visibilidad promedio, velocidad del viento promedio, precipitación y eventos durante el día (Por ejemplo, niebla, asoleamiento)

En (Wang and Kim, 2018), se probaron modelos basados en árboles de decisión y de redes neuronales, para pronosticar el nivel de disponibilidad de los sistemas *bike-sharing*. Basaron su modelo en (Wang, 2016) donde la demanda regional de renta de bicicletas se predecía usando *Machine learning*, específicamente, los modelos basados en árboles de decisión y en redes neuronales. Este modelo no consideraba factores como el clima, día de la semana u hora del día. (Datta, 2014) utilizó árboles de decisión, Random Forests y Adaboost para comparar los modelos Poisson empleados por *Data Science for Social Good* (DSSG). Esta tesis se enfoca en la variable tiempo que, a su vez, la divide en cuatro variables adicionales: hora del día, día de la semana, día del mes y mes. Además, considera temperatura, precipitación y nubosidad como variables climáticas.

Todos los estudios enfatizan la importancia de dividir el tiempo en varias variables (minutos, horas, días, día de la semana, etc.) y consideran importantes, las variables climáticas debido al alto

impacto que pueden tener sobre la demanda en el sistema. (Ashqar et al. 2014) y (Wang and Kim, 2018) compararon modelos basados en árboles de decisión y se concluyó que *Random Forests* es un mejor modelo para este problema, por tener una mejor adaptabilidad a los datos y una alta eficiencia computacional.

Esta investigación se enfoca en el entendimiento de las variables que impactan a los patrones de demanda del sistema *bike-sharing*, y a su respectiva predicción, en particular utilizando dos herramientas de *Machine learning*: Random Forests y XGBoost, siguiendo una metodología de ciencia de datos en diez pasos, la cuál es descrita detalladamente en el capítulo de Metodología. El modelo propuesto predice el flujo de viajes en bicicleta entre estaciones, para así, gráficamente comprender y analizar los datos de forma sencilla.

Capítulo 2

Aprendizaje automático (Machine learning)

Introducción

Es una forma de Inteligencia Artificial que habilita a un sistema a interpretar datos y usa una variedad de algoritmos que aprenden iterativamente a través de los datos para mejorar, describir y predecir resultados (Hurwitz and Kirsch, 2018). Sencillamente, *Machine learning* es un proceso automático para descubrir patrones y tendencias en datos que no pueden obtenerse mediante un simple análisis (Ghatak, 2017).

Machine learning transforma un pequeño conjunto de datos recopilados en un gran cúmulo de conocimiento aplicable; mientras más datos se le proporcionen al modelo, este será más preciso y se obtendrán resultados más significativos.

Fundamentalmente, la iteración es la esencia de *Machine learning* (Ghatak, 2017). A diferencia de la mayoría de los algoritmos, *Machine learning* no requiere de un programador para iniciar la inserción de algoritmos, si no que los mismos datos crean un modelo (Hurwitz and Kirsch, 2018).

Todos los algoritmos *Machine learning* utilizan datos de "entrenamiento" para obtener un modelo que es evaluado con un conjunto de datos no observados conocido como conjunto de prueba. También existen enfoques de *Machine learning* basados en los datos insertados, procesos específicos para manejar el proceso de entrenamiento del modelo y otras numerosas características de algoritmos *Machine learning.* En las siguientes secciones se explicarán algunos de los conceptos más importantes de *Machine learning.*

Aprendizaje supervisado y no supervisado

Hay dos tipos de *Machine learning* de acuerdo con el tipo de inserción de datos, los cuales son:

- Aprendizaje supervisado (*Supervised learning*)
- Aprendizaje no supervisado (*Unsupervised learning*)

Aprendizaje supervisado (Supervised learning)

(Ghatak, 2017) define al aprendizaje supervisado como un algoritmo que adquiere experiencia de un conjunto de datos que además de características contiene una variable objetivo, la cual es una función de las características.

En (James et al., 2013), el aprendizaje supervisado se define como, para cada observación de las medidas del predictor $x_i \ i = 1, \dots, n$, hay una respuesta de medida asociada y_i. Así, el objetivo es adaptar un modelo que relacione la respuesta con los predictores procurando predecir con precisión la respuesta a futuras observaciones (predicción) o bien, buscar un mejor entendimiento de la relación entre respuesta y predictor (inferencia).

A modo de ejemplo del aprendizaje supervisado está la predicción de demanda de un producto en distintos lugares dadas ciertas características tales como población, ubicación, clima de la región, ingreso promedio por ciudadano, etc. Otro ejemplo sería el diagnóstico de un paciente de acuerdo con los síntomas que presenta, es decir, se podría clasificar si un paciente está contagiado por COVID-19 según una observación de sus síntomas, sin ningún diagnóstico médico.

Aprendizaje no supervisado (Unsupervised learning)

En (Ghatak, 2017), el enfoque de aprendizaje no supervisado se define como un algoritmo que adquiere experiencia de un conjunto de datos que solo contiene características sin ninguna variable dependiente. Los algoritmos de este enfoque seccionan los datos en grupos de características.

En contraste, el aprendizaje no supervisado describe la situación en la que cada observación no tiene una respuesta asociada. En este escenario, se busca una relación entre los predictores o las observaciones. Por lo tanto, no es posible adaptar un modelo de regresión lineal debido a que no hay una respuesta variable. Una de las herramientas de *Machine learning* de aprendizaje no supervisado más populares es el de análisis de agrupamiento o de agrupación (*Clustering analysis*) (James et al., 2013).

En estas situaciones donde el análisis de datos no tiene una noción del contexto de los datos y no es posible etiquetarlos, se utiliza el aprendizaje no supervisado antes de un proceso de aprendizaje

supervisado. En ausencia de una variable objetivo, el proceso de aprendizaje no supervisado agrega etiquetas a los datos, así convirtiéndose en un aprendizaje supervisado (Hurwitz and Kirsch, 2018). Para un mejor entendimiento (James et al., 2013) describe el ejemplo siguiente para el análisis de agrupación (*Clustering*). En un estudio de segmentación de mercado podríamos observar características de clientes tal como su código postal, ingreso familiar, y hábitos de compra, sin embargo, no conocemos los hábitos de consumo, que sería la variable objetivo. Por lo tanto, con las características observadas podemos pensar en dos categorías para los clientes, de alto y bajo consumo. Con aprendizaje no supervisado es posible agrupar a los clientes con base a las variables medidas.

Debido a que las variables pueden ser categorizadas como cuantitativas y cualitativas, en *Machine learning* tiende a referirse como Regresión a aquellos problemas con una respuesta cuantitativa, y como Clasificación a los problemas donde la variable objetivo es cualitativa. A continuación, se describen brevemente ambos conceptos.

Regresión

En regresión, el algoritmo realiza la tarea de predecir un valor numérico para una inserción de datos. En regresión, se usa una función genérica $y = f(x)$ definida por el algoritmo y utilizando un vector X ingresado para predecir un resultado de salida numérico. (Ghatak, 2017).

Clasificación

En clasificación, el algoritmo realiza una especificación de cuál de las clases *k* pertenece a un dato ingresado. Si el algoritmo se define por una función genérica $y = f(x)$, la tarea de clasificación es tomar un dato ingresado definido por un vector X, de manera que el algoritmo asigne una categoría de clase *k* como el resultado de la función *f*, identificada como y. Un resultado numérico de la función *f* puede ser una distribución de probabilidad, asignando una probabilidad a las clases *k* predichas.

Datos de prueba y entrenamiento

Es esencial para el proceso de *Machine learning* dividir los datos totales en dos grupos, de prueba y de entrenamiento. Básicamente, los datos de entrenamiento es un conjunto de datos que

construyen al modelo, para luego, ser evaluado por un conjunto de datos el cuál es nombrado datos de prueba, los cuales son desconocidos para el modelo.

No existe una regla para la división de datos, no obstante, algunos porcentajes de división descritos por (Brownlee, 2020) pueden ser:

- Entrenamiento 80%, Prueba 20%
- Entrenamiento 70%, Prueba 30%
- Entrenamiento 67%, Prueba 33%
- Entrenamiento 50%, Prueba 50%

El error obtenido por el algoritmo usando los datos de entrenamiento es el error de entrenamiento del modelo. Por otra parte, el error de prueba es el error obtenido tras haber evaluado el modelo de entrenamiento con los datos de prueba. Es importante mencionar que el error de prueba es siempre mayor o igual que el error de entrenamiento, ya que los coeficientes del modelo son adaptados a los datos de entrenamiento. Asimismo, el error de entrenamiento es menor que el error de prueba, debido a que el error de prueba se obtiene al evaluar el modelo con datos no observados, por lo que se obtendrá un error mayor. Finalmente, el error de entrenamiento puede ser utilizado como punto de referencia para el error del modelo (Ghatak, 2017).

Si el volumen de datos recolectados es pequeño, se podría obtener un modelo deficiente para el problema, o bien, si no hay suficientes datos para aplicar un modelo de Machine learning, se podría implementar una validación cruzada con el objetivo de reducir el error del modelo y obtener mejores predicciones, este concepto se explicará en mayor detalle en la siguiente sección.

Validación cruzada (*Cross validation*) o métodos de remuestreo

(James et al., 2013) define a los métodos de remuestro como, aquellos métodos que involucran la creación repetida de muestras para un conjunto de entrenamiento y ajustar un modelo para cada muestra realizada. De esta forma, obtener mejores resultados del modelo construido, o conseguir información adicional importante para el modelo.

Los métodos de remuestreo son utilizados para mejorar el desempeño del modelo de Machine learning, como, por ejemplo, al proteger el modelo del sobreajuste (Véase en la página 11, para mayor explicación) de los datos de entrenamiento (Mujtaba, 2020).

Estos métodos utilizan muchos recursos computacionales ya que, el proceso implica ajustar un modelo de *Machine learning* múltiples veces, utilizando diferentes subconjuntos de datos de entrenamiento, para luego evaluar el modelo obtenido con el conjunto de datos de prueba (James et al., 2013). Esencialmente, hay tres tipos de métodos de remuestreo, los cuales son descritos a continuación:

1. Hold out

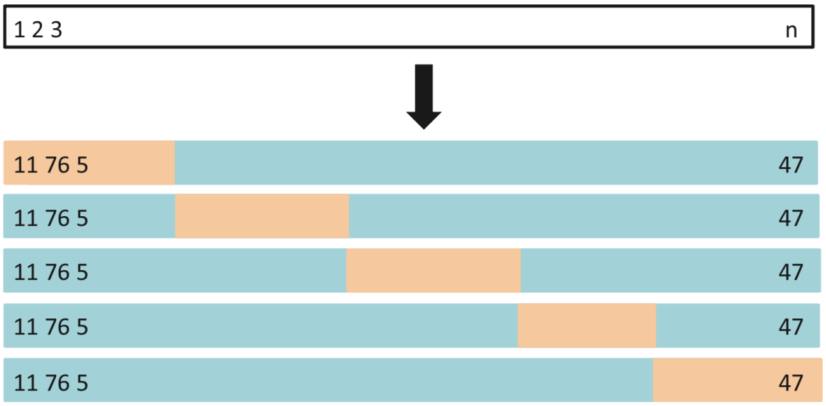
(James et al., 2013) expone que este método es el más básico y común entre los métodos de remuestreo, porque simplemente divide el conjunto de datos en dos partes, los datos de entrenamiento y los datos de prueba. Fundamentalmente, el proceso *Hold-out*, involucra dividir de manera aleatoria el conjunto de datos en dos partes, para luego ajustar el modelo en el conjunto de datos de entrenamiento, y el modelo obtenido de este proceso, es examinado con los datos de prueba. Como se ha mencionado anteriormente en este capítulo, el tamaño del conjunto de datos es dividida en 80% para los datos de entrenamiento y el 20% para los datos de prueba, o en porcentajes con valores similares (Véase en la sección de Datos de prueba y entrenamiento, página 10).

2. Validación cruzado K-fold (K-fold cross-validation)

De acuerdo con (Mujtaba, 2020), la validación cruzada *K-fold* es una forma de mejorar el desempeño del método *Hold-out*, ya que el modelo ajustado no sólo depende de una división del conjunto de datos, por lo contrario, este método implica dividir los datos en *k-folds* (k-divisiones), que consiste en un número de subconjuntos *k*, de datos con un mismo tamaño. De todos los subconjuntos *k*, los primeros subconjuntos *k-1* son utilizados para el entrenamiento del modelo y así, el subconjunto restante es utilizado para poner a prueba el modelo. Este proceso debe ser repetido *k* veces, y el error de cada *k* división será obtenido y promediado para tener el error total medio del modelo (Ghatak, 2017).

Para un mejor entendimiento de este proceso, la *Figura 2.1* muestra un proceso de validación cruzado de 5 divisiones, indicando el conjunto de datos total en la parte superior de la figura, y debajo de esto, se presenta las 5 divisiones en dos colores, beige para las muestras de datos de prueba y en azul, las muestras de datos de entrenamiento.

Figura 2.1. Representación de un proceso de validación cruzada de 5 divisiones



Recuperado: (James et al., 2013)

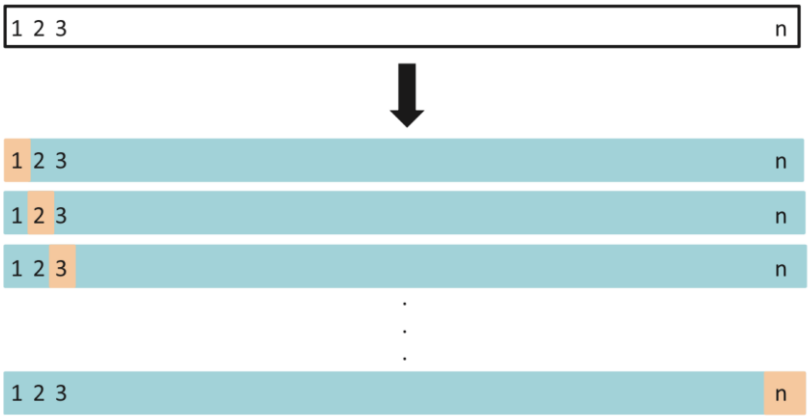
3. Leave one out cross-validation (LOOCV)

Leave one out cross-validation (LOOCV) es un método de remuestreo que está muy relacionado al método de *Hold-out* debido a que LOOCV involucra dividir el conjunto de datos, en dos partes, datos de entrenamiento y prueba. En lugar de dividir el conjunto de datos en partes iguales, como lo hace el método de validación cruzada *k-fold*, LOOCV solo deja una observación para probar el modelo entrenado con el resto de las observaciones (*n-1*). Es decir, una observación (x_1, y_1) es utilizada para probar el modelo y, las observaciones restantes $\{(x_2, y_2), \ldots, (x_n, y_n)\}$ construyen el modelo. Este proceso es repetido hasta que un número *n* de modelos hayan sido construidos. Como se podrá inferir, este proceso es exhaustivo y requiere muchos recursos computacionales, por lo tanto, este método de validación cruzada es utilizado en situaciones donde el conjunto de datos total es muy pequeño, de lo contrario, se recomendaría utilizar otros métodos como *Hold-out* y validación cruzada *k-folds* (Ghatak, 2017).

En la Figura 2.2 se puede observar la representación del proceso de division de los datos para el LOOCV. El conjunto de datos totales se observa en la parte superior de la imagen, y debajo se aprecia en color beige, el dato que se usará para evaluar el modelo construído con el resto de los

datos, que son mostrados en azul. Los números en cada recuadro describen el número de la observación del conjunto de datos, por lo que se puede determiner que el proceso tendrá *n* procesos de contrucción y evaluación del modelo.

Figura 2.2. Representación del proceso del método LOOCV



Un conjunto de datos n son repetidamente divididos en subconjuntos de datos de entrenamiento (Mostrado en azul), conteniendo todos los datos excepto uno, y el subconjunto de prueba conteniendo solo la observación restante (Mostrada en beige). El error de prueba es estimado al promediar los n resultados de MSE. El primer subconjunto de entrenamiento tiene a todos los datos excepto al primero, el segundo subconjunto de entrenamiento tiene a todos los datos, excepto al segundo dato, y así sucesivamente.

Recuperado: (James et al., 2013)

Compensación de sesgo-varianza

Antes de explicar qué es la compensación sesgo-varianza, es importante definir dos conceptos:

- Varianza
- Sesgo

Según (James et al., 2013), la varianza se refiere a la cantidad por la cual $\hat{f}$ (Función estimada) cambiaría si se estima utilizando un conjunto de entrenamiento distinto. Idealmente la estimación de f no debería variar mucho entre los conjuntos de datos de entrenamiento. Sin embargo, si un modelo tiene una varianza alta, entonces pequeños cambios en un conjunto de datos de entrenamiento, resultarán en grandes cambios para $\hat{f}$.

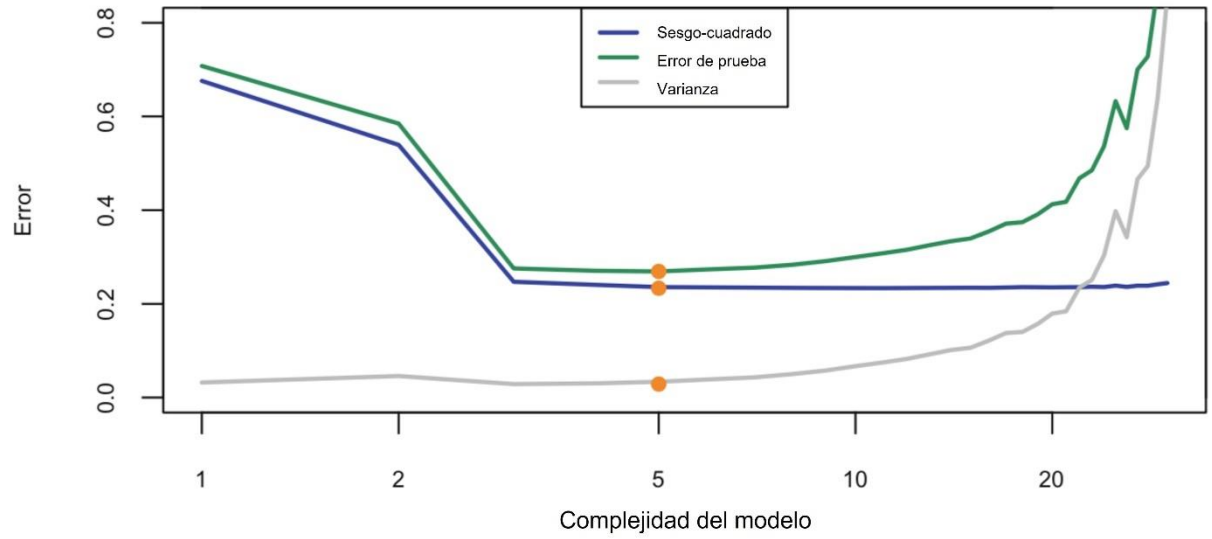
Por otro lado, sesgo, es definido por (James et., 2013) como el error que es obtenido al modelar un problema complejo de la vida real con un modelo muy sencillo. Por ejemplo, se ajusta una regresión lineal a un problema de la vida real, que es muy probable a que no siga un comportamiento lineal simple, por lo que resultaría en un sesgo para la estimación de f.

Estos dos conceptos podrían describirse, para Machine learning, como:

- El sesgo disminuye con un modelo complejo
- La varianza aumenta con un modelo complejo

Cabe recalcar que, la compensación sesgo-varianza está muy relacionada al error cuadrático medio (*Root mean squared error*, RMSE), una métrica descrita en la sección de Evaluación del capítulo de Metodología, página 37), ya que el índice de cambio del sesgo y la varianza determina si el RMSE de prueba incrementa o disminuye. Por lo cual, si incrementa la flexibilidad del modelo, es decir, se crea un modelo más complejo, inicialmente el sesgo del modelo tiende a disminuir y la varianza a incrementar, por consecuencia el RMSE de prueba, baja. Sin embargo, en algún punto del proceso el incrementar el número de variables en el modelo, no tendrá impacto en el sesgo y la varianza incrementará significativamente, así como es descrito en (James et al., 2013). Por consiguiente, la compensación de sesgo-varianza se refiere a encontrar un punto específico donde el sesgo y la varianza tienen el error mínimo alcanzable, como se presenta en la *Figura 2.3*, donde se muestra con un punto naranja, la complejidad correcta del modelo, donde el sesgo ya no disminuye notablemente y la varianza no aumenta dramáticamente.

Figura 2.3. Compensación de sesgo-varianza



Recuperado: (Ghatak, 2017)

Cualquier modelo que contemplara una complejidad antes del punto naranja (Complejidad correcta del modelo), resultaría en un modelo subajustado, y en caso contrario, si la complejidad del modelo fuera seleccionada con valores superiores a los del punto naranja, el modelo se sobreajustaría (Ghatak, 2017). Estas dos definiciones, subajuste y sobreajuste del modelo serán descritas en la siguiente sección.

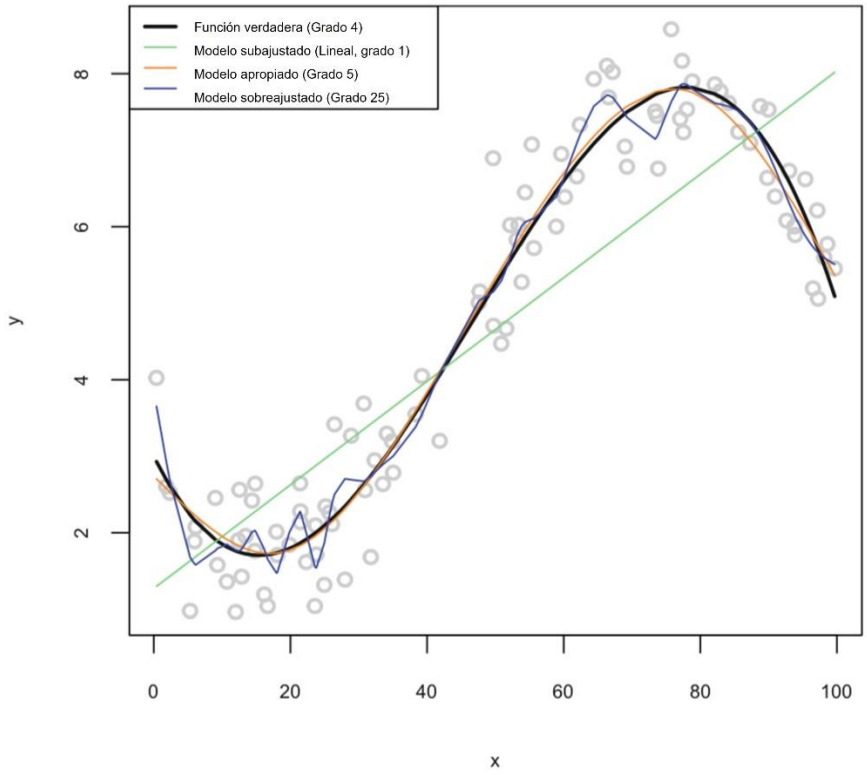
Subajuste y sobreajuste (*Underfitting* and *overfitting*)

En *Machine learning* es muy común decir que un modelo está subajustado o sobreajustado a los datos. Básicamente, el subajuste ocurre cuando el modelo no captura el comportamiento real de los datos, y, por otro lado, el sobreajuste se presenta cuando el modelo es creado para un conjunto de datos en específico y el modelo es muy complejo, por lo tanto, el modelo tendrá una capacidad baja de representar el comportamiento real de los datos, es decir, cuando un modelo está sobreajustado, se refiere a que éste describe el ruido de los datos.

Como puede ser observado en la *Figura 2.4*, una función polinomial de grado 4 (con ruido agregado) representa una función verdadera (Línea negra). Y como un ejemplo de un modelo subajustado que trata de describir la función real con una simple regresión lineal (Línea verde). Por el contrario, una función polinomial de grado 25 (Línea azul) está describiendo la función real

y, además, el ruido de esta, eso quiere decir que es un modelo sobreajustado. Un modelo apropiado para esta función podría ser un polinomio de grado 5 mostrado en la gráfica con una línea naranja.

Figura 2.4. Representación del subajuste y sobreajuste



Recuperado: (Ghatak, 2017)

Hiperparámetros

Los hiperparámetros son parámetros intrínsecos para el modelo, también son usados para controlar el modelaje de este. Los hiperparámetros no aprenden de los datos, pero sí causan efectos en el desempeño del modelo a partir de la modificación del proceso de su creación. Algunos de los hiperparámetros son:

- Índice de aprendizaje del algoritmo
- Número de árboles
- Número de hojas o profundidad del árbol
- Número de centroides en clustering
- Número de variables de muestro aleatorio

Los hiperparámetros enlistados son sólo algunos de la gran cantidad de hiperparámetros existentes.

Un importante concepto es el de optimización de hiperparámetros (*Hyperparameter tuning*), el cual es el proceso de obtener los mejores valores de los hiperparámetros para un problema específico, esto se consigue a partir de un proceso iterativo, donde el modelo de entrenamiento pone a prueba el desempeño de un conjunto de valores de los hiperparámetros para medir y conseguir el mejor valor para el modelo.

Los hiperparámetros que fueron modificados en el modelo de predicción de esta investigación son explicados con mayor detalle en la sección de Metodología (Página 35-36).

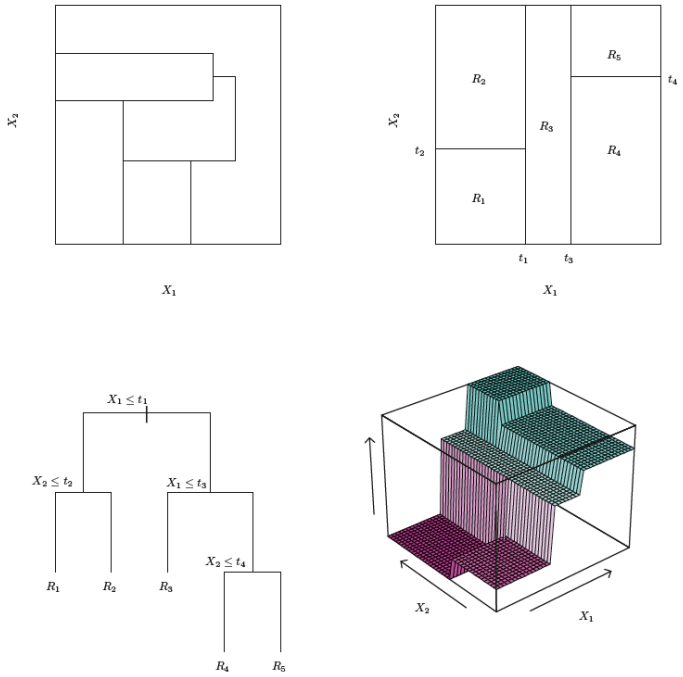
Todos los conceptos descritos en este capítulo son muy importantes para la comprensión de los algoritmos de *Machine learning* implementados para la predicción de la demanda del sistema *bike-sharing*. Es muy importante mencionar que en esta investigación se utilizó un aprendizaje supervisado de regresión debido a que se tiene un conjunto de variables independientes o predictores y solo un resultado asociado o una variable dependiente.

De acuerdo con pruebas realizadas en esta investigación y artículos científicos relacionados a análisis de BSS, los modelos basados en árboles de decisión son uno de los mejores enfoques para la predicción de la demanda de estos sistemas. Es por esto que, los modelos basados en árboles de decisión son explicados en la siguiente sección.

Modelos basados en árboles de decisión

Estos métodos son algoritmos que involucran estratificación o segmentación del espacio predictor a un número de regiones simple como puede observarse en la *Figura 2.5*. Básicamente, el algoritmo realiza una predicción para una observación dada, utilizando el promedio o la moda de las observaciones de entrenamiento en la región a la que pertenecen. Gráficamente, las reglas que segmentan al espacio predictor son representadas como un árbol *(Obsérvese parte inferior izquierda en Figura 2.5)*, por esta razón se conocen como árboles de decisión.

Figura 2.5. Segmentando el espacio predictor.



Arriba a la izquierda: Una división de un espacio bidimensional que no podría ser obtenida de una división recursiva binaria. Arriba a la derecha: El resultado de una división recursiva binaria sobre un espacio bidimensional. Abajo a la izquierda: El correspondiente árbol de la partición del panel superior. Abajo a la derecha: Una perspectiva del gráfico de la superficie de predicción correspondiente al árbol de la parte superior.

Recuperado: (James et al., 2013)

(James et al., 2013) define una forma sencilla de construir un árbol de regresión basado en dos pasos:

1. Dividir el espacio predictor -esto es, el conjunto de valores posibles para $X_1, X_2, \ldots X_p$- entre regiones *J* distintas y no sobrepuestas $R_1, R_2, \ldots R_j$.

2. Para cada observación encontrada en la región R_j, se realiza la misma predicción, la cual es simplemente el promedio de valores de respuesta para las observaciones de entrenamiento en R_j.

Generalmente, los árboles de decisión no tienen un nivel alto de precisión en comparación a algoritmos más complejos. Sin embargo, al agregar muchos árboles de decisión, el desempeño de

la predicción incrementa significativamente, utilizando métodos como *bagging*, *boosting* y Random Forests (James et al., 2013).

Estos algoritmos son considerados como algoritmos de aprendizaje ensamblado ya que, estos mejoran su desempeño al combinar o ensamblar múltiples modelos.

Pese a la amplia variedad de métodos basados en árboles de decisión, para esta tesis se consideraron dos de ellos: Random Forests (RF) y eXtreme Gradient *Boosting* (XGBoost). Este último del tipo *Boosting*, que consiste en la combinación de algoritmos de *Machine learning*. Ambos se explicarán con más detalle en la siguiente sección.

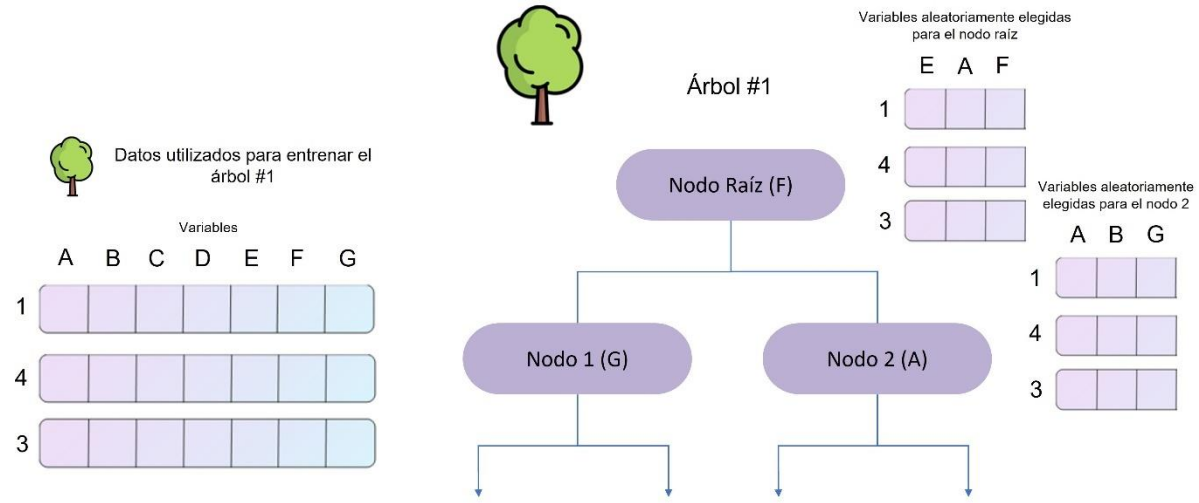
Random Forests (RF)

Como se mencionó anteriormente, Random Forests es un algoritmo basado en árboles de decisión. Este es un modelo de ensamblado, usando muchos "aprendices débiles", es decir, árboles de decisión simples y construir un "aprendiz fuerte". En otras palabras, Random Forests está construido con un gran número de pequeños árboles de decisión. Con cada predicción realizada por cada uno de ellos, Random Forests produce una mejor predicción que un árbol de decisión simple (Wood, 2021).

Este es una combinación del proceso *bagging* y métodos de enfoque aleatorios, una fusión única que da origen a este algoritmo. *Bagging* es parte del proceso de Random Forests al permitir que cada árbol de decisión que lo conforma sea entrenado por una muestra con reemplazo, es decir, que un dato puede presentarse en la misma muestra más de una vez, resultando en diferentes árboles (Yiu, 2019).

Random Forests incluye aleatoriedad en cada una de sus ramificaciones, al seleccionar una variable de un subconjunto del total de variables que fue creado de manera aleatoria, como se muestra en la *Figura 2.6*. En el nodo 1, la variable G fue seleccionada de un subconjunto de variables (C, G, D) que asimismo fueron aleatoriamente seleccionadas de un total de variables (A, B, C, D, E, F, G).

Figura 2.6. Representación de la selección aleatoria de variables.

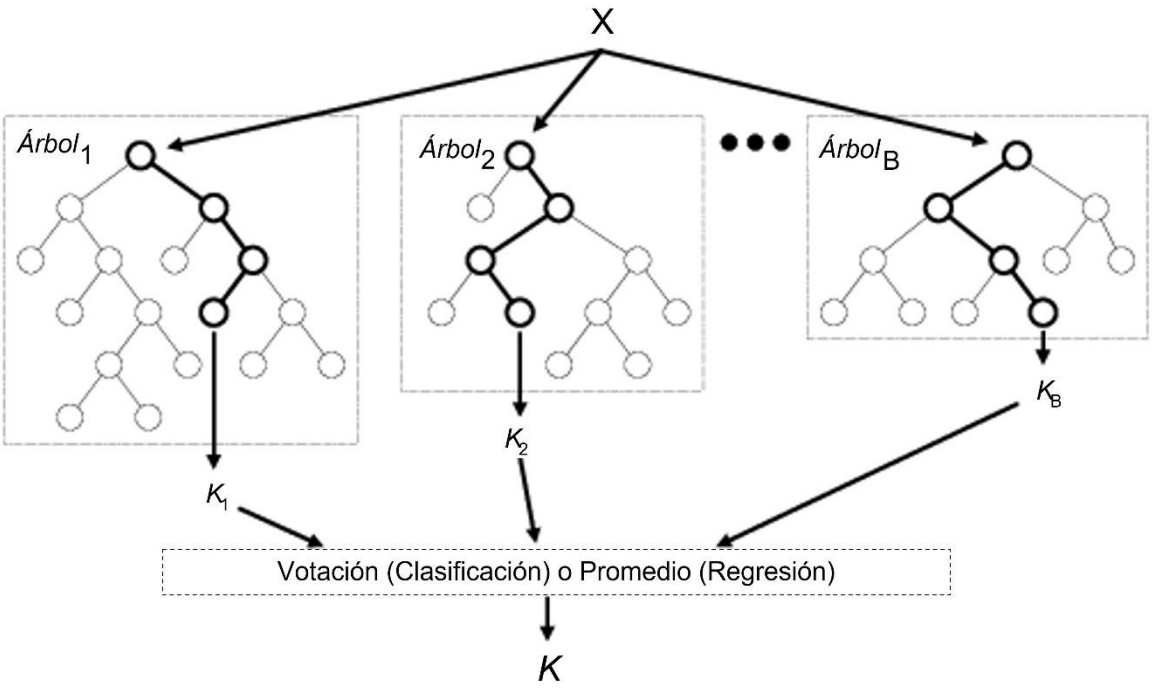


Recuperado: (Anónimo, 2020)

(Anónimo, 2020) afirma que Random Forest genera una mejora significativa sobre los árboles de decision, mediante la aplicación de dos conceptos (*Bagging* y Aleatoriedad), ya que, decorrelaciona el grupo de árboles de decisión construidos. Es decir, Random Forests provoca una divergencia de los árboles por medio de la asignación de un subconjunto de predictores para cada árbol, evitando de esta manera la correlación entre todos los árboles de decisión. Matemáticamente hablando, el promedio de ciertas cantidades reduce la varianza, así mejorando la predicción utilizando Random Forests (James et al., 2013).

Una representación simple del trabajo interno de Random Forests para clasificación y regresión se muestra en la *Figura 2.7*, donde los resultados de cada árbol creado son promediados para la obtención del resultado final en los modelos de regresión y, en modelos de clasificación, el resultado final es obtenido por medio de un proceso de votación.

Figura 2.7. Representación del proceso interno del algoritmo de Random Forests.



Recuperado: (Verikas et al., 2016)

Algunas de las ventajas de esta herramienta de *Machine learning* son:

- Fácil aplicación
- Uso relativamente bajo de recursos de cómputo
- Buen desempeño al manipular grandes cantidades de datos con dimensionalidad alta y tipos de variables heterogéneas (Muchas variables, variables cuantitativas y cualitativas en el mismo conjunto de datos, según (Wood, 2021)).

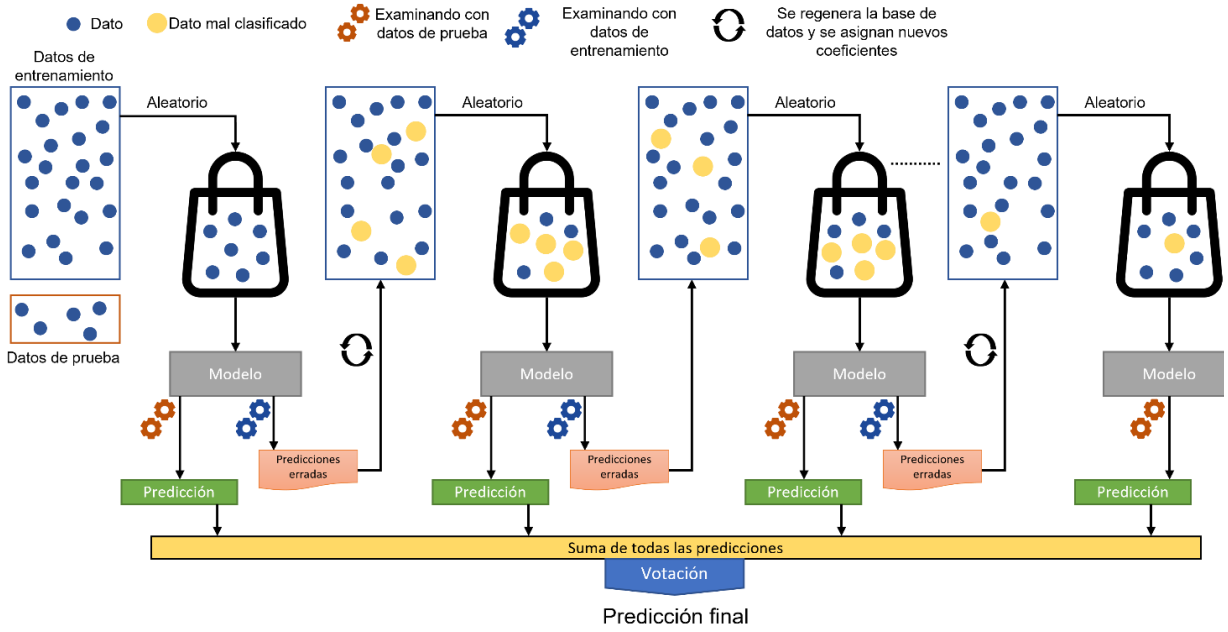
Sin embargo, una de las mayores debilidades de Random Forests es la interpretabilidad, ya que los árboles de decisión pueden ser fácilmente interpretados de manera gráfica, por el contrario, Random Forests es mucho más complejo y casi imposible de interpretar.

Para una explicación a detalle sobre Random Forests, ver en (James et al., 2013)

eXtreme Gradient Boosting (XGBoost)

Para poder describir esta herramienta de *Machine learning*, primero habrá que definir el término *Boosting*. En *Machine learning*, *Boosting* se refiere a una técnica de aprendizaje de ensamble secuencial. Esto es, que las decisiones de múltiples algoritmos de *Machine learning* se combinan para reducir errores y mejorar las predicciones a través de una secuencia de modelos de *Machine learning* creados a partir de datos provistos (Malik et al., 2020). Esta técnica se describe gráficamente en la *Figura 2.8*.

Figura 2.8. Representación del trabajo interno de un algoritmo *Boosting*



Recuperado: (Malik et al., 2020)

XGBoost es un modelo especial de *Gradient Boosting*. En este modelo *Boosting* los errores son minimizados a través de un algoritmo de gradiente descendiente. XGBoost asigna coeficientes por medio de un gradiente en dirección de una función de pérdida, optimizando la pérdida del modelo actualizando los coeficientes. Usualmente, la pérdida del modelo se define como la diferencia entre los valores predichos y los valores reales (Malik et al., 2020). Llevando a cabo este proceso, el modelo emplea un proceso aditivo por lo que un nuevo árbol de decisión es agregado, uno a la vez, lo cual minimiza las pérdidas con el gradiente descendiente. Con esta explicación, se podría

concluir que XGBoost es una versión mejorada de *Gradient Boosting* en cuanto a efectividad, eficiencia computacional, y desempeño del modelo.

Para más información de este modelo de *Machine learning*, véase (Malik et al., 2020).

Realizando una comparación entre Random Forests y XGBoost, Random Forests realiza un pormedio de los resultados de cada árbol de decisión, y a su vez, XGBoost asigna diferentes ponderaciones al resultado de cada árbol de decisión. Adicionalmente, XGBoost utiliza una única función de pérdida para entrenar al modelo, por esto, este algoritmo puede ser utilizado para cualquier tarea que pueda ser expresada a través de una función de pérdida. Además, XGBoost puede superar a Random Forests al poder realizar modelos más complejos y escalables, sin embargo, esto mismo hace que XGBoost sea un poco más complicado de adaptar a un sistema en comparación a Random Forests, en términos del proceso de optimizado de hiperparámetros (Wood, 2021).

En conclusión, los beneficios de aplicar algoritmos de XGBoost, son:

- Mejor desempeño que Random Forests
- Mayor flexibilidad
- Una importante reducción en el tiempo de entrenamiento

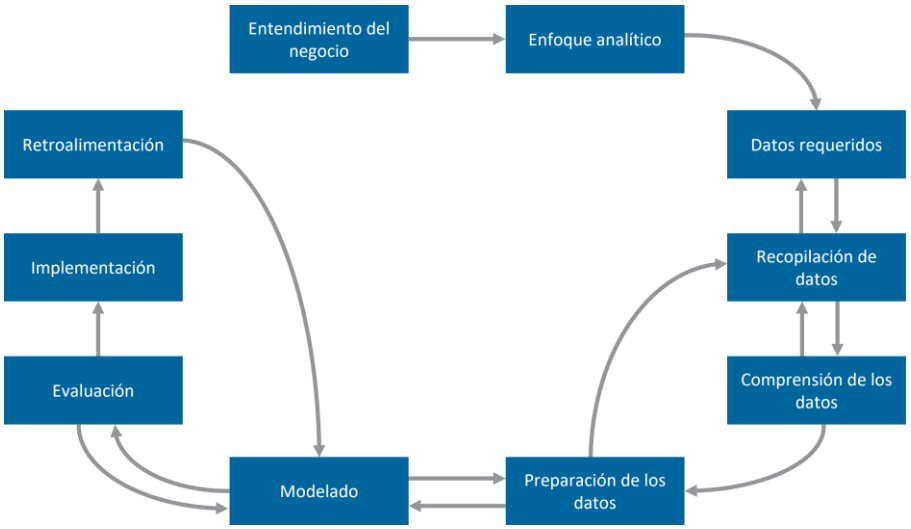
El ultimo beneficio enlistado es demostrado en la presente investigación, donde el tiempo de entrenamineto de XGBoost es tres veces más rápido que el tiempo de Random Forests, como se muestra en la *Tabla 4.4 (Véase página 53)*.

Capítulo 3

Metodología

Como se mencionó anteriormente, uno de los principales objetivos de esta investigación es explicar la construcción del modelo de predicción, cómo se descubrieron los patrones de demanda del sistema y qué variables se consideraron para el modelo. La metodología aplicada para alcanzar estos objetivos es la de ciencia de datos, tal como propone (Rollins, 2015) mostrado en el diagrama de diez pasos a continuación *(Figura 3.1)*:

Figura 3.1. Metodología fundamental para la ciencia de datos.



Recuperado: (Rollins, 2015)

Entendimiento del negocio (Entendimiento del sistema)

Mencionado en la sección de descripción del problema, Ernesto García Almaraz afirmó que no hay un conocimiento estructurado ni documentado de la caracterización de la demanda del sistema Bicipuma. A lo largo del tiempo que ha operado Bicipuma, los patrones de demanda se han aprendido de manera empírica, por lo que, en ocasiones, resulta en una respuesta ineficiente a la demanda del sistema y como consecuencia, es común observar escasez y excedente de bicicletas en algunas de las estaciones de Bicipuma, afectando directamente a los usuarios de este servicio.

Un objetivo de esta investigación es comprender las variables y factores que afectan la demanda de Bicipuma, también es entender el patrón de demanda entre estaciones para que la demanda del servicio pueda ser predicha. Con más de dos millones de viajes registrados por Bicipuma desde enero 2017 a noviembre 2019, y con datos climatológicos obtenidos para este mismo periodo de tiempo, estos objetivos pudieron ser alcanzados.

Enfoque analítico

Es determinante describir el resultado del modelo para poder comprender y definir cómo se optimizará el sistema, así como analizar los modelos que representen mejores soluciones a este, por lo tanto, en este documento se definió un enfoque de regresión para obtener el número de viajes realizados desde una estación a otra en un intervalo de tiempo determinado. Así, Bicipuma contando con 14 estaciones, puede ser considerado un sistema con una red con 14 nodos y 182 enlaces, es decir, el resultado del modelo de predicción es el número de viajes por cada enlace de la red, presentándose de una manera entendible y más fácil de manejar para futuras investigaciones.

Los modelos de regresión de *Machine learning* utilizados en esta investigación (Random Forests y XGBoost) fueron seleccionados por su adaptabilidad a este estudio en específico. Además, ambos modelos frecuentemente son seleccionados en otros artículos científicos relacionados a este tema como (Yang et al., 2020), (Wang and Kim, 2018), (Datta, 2014) y (Ashqar et al., 2019) y competencias de Kaggle de predicciones sobre *BSS*, entre otras investigaciones.

Datos requeridos

Para poder obtener el resultado deseado, es decir, obtener el número de viajes realizados desde una estación a otra en un intervalo de tiempo determinado para cada uno de los enlaces de la red, por lo menos es necesario los siguientes datos:

- Fecha
- Hora de solicitud de la bicicleta
- Hora de entrega de la bicicleta
- Estación de origen
- Estación de destino

Esta información es suficiente para analizar, obtener puntos clave para la comprensión del sistema y descubrir el patrón de comportamiento de este.

Recopilación de datos

Los datos de Bicipuma son registrados a través de un sistema de código de barras. El personal de este servicio escanea el código de la bicicleta y de la identificación del usuario, relacionando estos dos elementos del sistema y registrando cada viaje realizado durante la operación de Bicipuma para poder tener una trazabilidad de los usuarios.

El coordinador del programa Bicipuma proporcionó la base de datos requerida para desarrollar esta investigación, los cuales consisten en cada registro de viaje realizado desde enero 2017 a noviembre 2019, conteniendo la información de cada viaje como fecha, hora de solicitud, hora de entrega, número de identificación de la bicicleta, estación de origen, estación de destino y número de identificación del usuario, este último fue omitido por el personal de Bicipuma para asegurar la privacidad de los usuarios de este servicio. El total de los datos recolectados superan los dos millones de viajes. Específicamente, estos datos fueron proporcionados a través de tres archivos de Excel con el formato presentado en la tabla siguiente *(Tabla 3.1)*.

Tabla 3.1
Formato de la base de datos de Bicipuma

Fecha	Tiempo_solicitud	Tiempo_entrega	ID_bicicleta	Origen	Destino
'2019-11-14'	'06:53:14'	'07:00:10'	4324	2	9
'2019-11-14'	'10:03:14'	'10:12:08'	1748	6	6
'2019-11-14'	'10:12:26'	'10:22:34'	1748	6	11

Recuperado: Elaboración propia

Adicionalmente, los datos climatológicos fueron extraídos desde la base de datos del Instituto Meteorológico de la UNAM, en 36 archivos de Excel en formato CSV.

Tanto los datos climatológicos como los registros de viaje de Bicipuma fueron comparados en los mismos períodos de tiempo. La base de datos climatológica incluye información tal como, registro

de hora, temperatura promedio, humedad relativa promedio, velocidad del viento promedio, promedio de dirección del viento, desviación estándar de la dirección del viento, velocidad del viento máxima, lluvia total, presión atmosférica promedio, radiación promedio, y visibilidad promedio.

El observatorio atmosférico de la UNAM se encuentra dentro del campus (con coordenadas 19.3262° N and 99.1761° W), tal como el sistema *bike-sharing* estudiado en esta tesis. Dicho observatorio registra datos por minuto las 24 horas del día.

Comprensión de los datos

Un paso fundamental en el proceso de ciencia de datos es, entender los datos proporcionados y comprenderlos con un enfoque sistémico del servicio Bicipuma, esto significa que, considerar cada factor que impacta al sistema, entender el comportamiento de los usuarios el cual afecta directamente en la predicción de la demanda de Bicipuma, esto es un proceso fundamental para los siguientes pasos de la metodología empleada.

Los datos proporcionados y la información recabada en una entrevista con el coordinador general del sistema Bicipuma, fue relevante notar algunas inconsistencias o errores de registro en la base de datos tales como que, el sistema Bicipuma tiene un horario de operación específico, pero algunos de los registros podrían estar fuera de este horario, registros de viaje duplicados, problemas con la hora registrada o un registro teniendo la misma estación como origen y destino, suponiendo que el usuario utilizó el servicio para propósitos recreativos, es decir, solicitó una bicicleta en la estación 1 y volvió a la misma. Estos y otros detalles e información importante fueron considerados para esta investigación, y fue determinante para la etapa de preparación de los datos.

Preparación de los datos

Así como en cada proyecto de ciencia de datos, uno de los pasos que más requiere tiempo es la limpieza y preparación de los datos. En este proceso, es común enfrentarnos con problemas de datos faltantes, inválidos, observaciones duplicadas y ajustar los datos en el formato deseado para su mejor manejo.

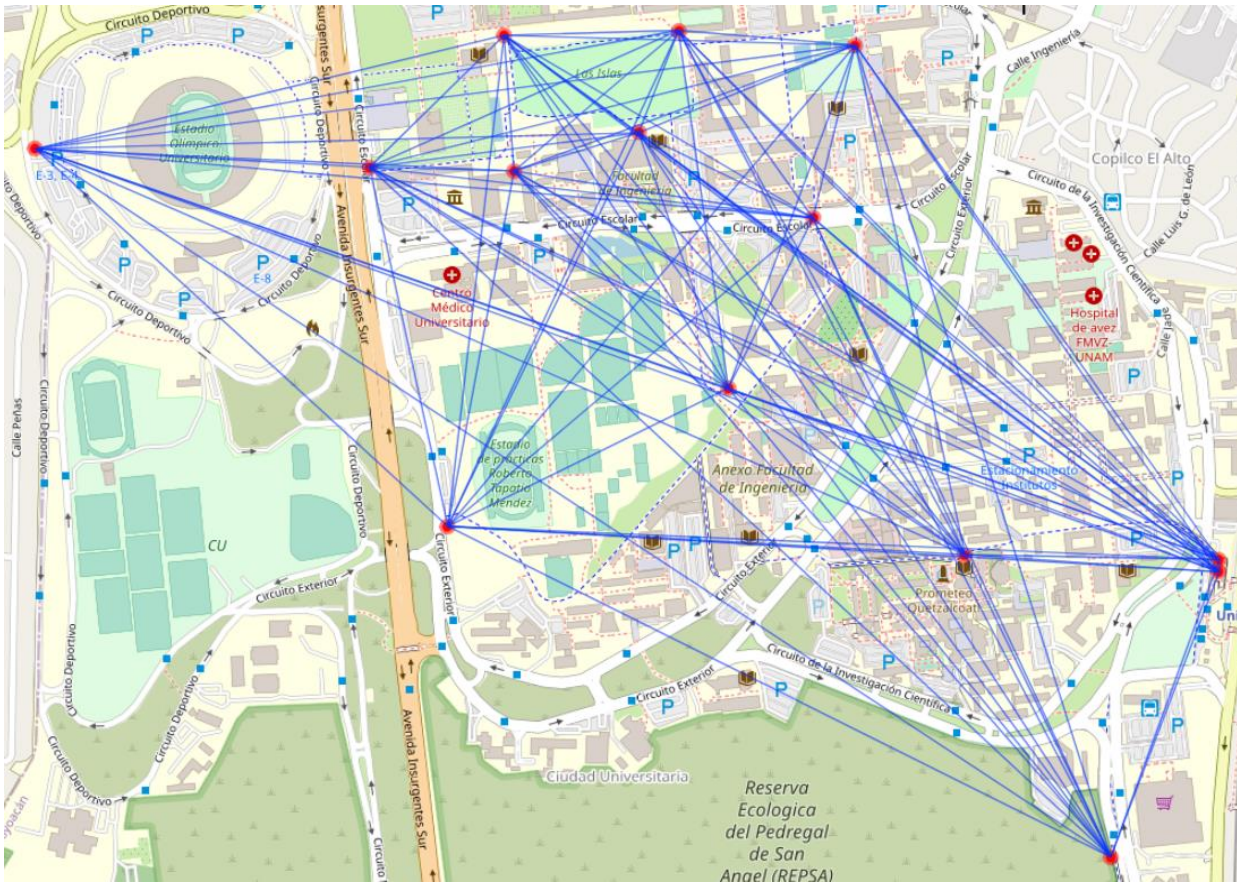
Como primer paso de la preparación de los datos, la base de datos de Bicipuma y la base de datos climatológicos fueron unidas, resultando en una base de datos general, la cual contiene todos los viajes realizados con sus correspondientes datos climatológicos. Lo anterior, fue realizado para efectuar un análisis más profundo sobre la relación entre las variables climatológicas y el número de viajes realizados en un periodo de tiempo específico.

Figura 3.2 Viaje en bicicleta por campus universitario utilizando Bicipuma.

Recuperado: (The observer, 2018).

Para esta investigación, los datos fueron agrupados por intervalos de tiempo de 30 minutos y realizando el conteo del número de viajes realizados en el sistema en ese intervalo. En consecuencia, fue añadida una nueva variable a la base de datos, la variable "N", que puede ser descrita como el número de viajes realizados en un intervalo de tiempo específico de 30 minutos. Si Bicipuma tiene 14 estaciones, entonces es considerado como una red de 14 nodos y 182 enlaces *(Ver Figura 3.3)*, y los datos fueron ajustados a esta configuración para realizar la predicción para cada enlace de la red y así, comprender y analizar el sistema, de una manera más simple.

Figura 3.3 Representación de la red Bicipuma con 14 nodos y 182 enlaces.



Mapa creado con el software R.

Recuperado: Elaboración propia.

Con el propósito de hacer una base de datos apropiada para la manipulación y el análisis de los datos para este problema, se realizaron las siguientes actividades, sólo mencionando algunas de las más relevantes:

- Agrupación de datos por intervalo de 30 minutos (El número de viajes fue sumado; los registros climatológicos fueron promediados)
- Ajustar una hora de diferencia desde 27 de octubre 2019 hasta 4 noviembre 2019 y, desde 11 marzo 2019 hasta 5 abril 2019
- Omitir los viajes duplicados
- Omitir viajes fuera del horario de operación
- Omitir viajes a la estación de origen

Las dos últimas actividades mencionadas fueron analizadas y se determinó que ninguna de ellas presentaba un patrón de demanda o comportamiento, ni tenía ninguna relación con alguna variable considerada en esta tesis, además de que el volumen de estas observaciones era bajo.

Con el proceso de depuración de datos terminado, se procedió a transformar la base de datos en, datos ordenados, (*Tidy data*, en inglés) en otras palabras, forzar a que cada columna de la base de datos sea una variable, cada fila sea una observación y que cada celda tenga un único valor (Wickham, 2014), esto para un análisis correcto de los datos recabados.

Modelado

La comprensión de la relación entre cada variable y el resultado del modelo es muy importante para el modelado del sistema, para poder seleccionar las variables que realmente tienen un impacto en la demanda del sistema y descartar las que no lo afectan. En esta sección, se analizan todas las variables que fueron consideradas en el modelo de predicción, dividiendo la sección en tres partes:

- Análisis de las variables climatológicas
- Análisis de las variables de tiempo
- Análisis de la variable dependiente o N

Análisis de variables climatológicas

Como se ha mencionado anteriormente, cada una de las variables climatológicas fue analizada individualmente a través de una gráfica. La observación y el análisis de cada gráfica, la información proporcionada por el personal, y la comprensión de la red Bicipuma como un sistema, fue muy importante para decidir cuáles de las variables formarían parte del modelo de predicción. Todas las variables climatológicas fueron individualmente graficadas contra el número de viajes realizados en un intervalo de 30 minutos, con tres diferentes enfoques o tipos de gráficas:

- Total de viajes realizados en un intervalo de 30 minutos vs. una variable climatológica
- Total de viajes realizados en un intervalo de 30 minutos vs. una variable climatológica, agrupado por semana
- Total de viajes realizados en un intervalo de 30 minutos vs. tiempo de operación al día, y en color, los valores de la variable climatológica

Según el análisis de estas gráficas, fue determinado si la correlación es significativa para cada variable climatológica respecto al número de viajes del sistema, para decidir si la variable debía ser mantenida o no, en el modelo de predicción.

Análisis de variables de tiempo

Tal como el análisis de variables climatológicas, la variable de tiempo pasó por el mismo proceso. Cada variable fue trazada en relación con N para cada enlace de la red Bicipuma, analizando el sistema, descubriendo cada patrón de viaje, y comprendiendo la importancia de la relación entre estas variables.

Después de un análisis sistémico breve, se determinó dividir la variable de tiempo en intervalos de 30 minutos, días de la semana, semana, mes, semestre y año, para así analizarlos y decidir cuáles suponen una importancia considerable en el modelo de predicción. Por lo tanto, todas estas variables fueron graficadas para analizar patrones y entender la relación de dichas variables con el total de viajes realizados en el sistema.

La siguiente lista presenta las variables analizadas y la definición de cada una de ellas para explicar todos los valores posibles que pudieron ser observados.

- Intervalos de 30 minutos – Número total de viajes en bicicleta realizados en intervalos específicos de 30 minutos dentro del horario operacional de Bicipuma. Es decir, hay 23 intervalos posibles, empezando por el intervalo entre 6:30 - 7:00 hasta 17:30 – 18:00. Cada variable es dicotómica con valores de 0 y 1, indicando que un viaje fue realizado dentro de un intervalo específico con 1, y con 0 si el viaje no fue realizado en este intervalo. Por lo que un viaje registrado puede solo tener valor 1 en un intervalo, el resto tendrá 0.
- Días de la semana – Los días de la semana se dividieron en 5 valores, de lunes a viernes. Cada variable es dicotómica, el valor 1 indica que el viaje se realizó en un día específico y 0 indicando lo contrario. Los fines de semana son omitidos puesto que Bicipuma no opera estos días.
- Semana – Esta variable indica el número de la semana del año, considerando valores de 1 a 52.
- Mes – Indica el mes de los viajes, tomando valores de 1 a 12, esto es, enero a diciembre.

- Semestre – Esta variable tiene dos valores posibles, 1 para el período enero-junio, y 2 para julio-diciembre.
- Año – Esta variable toma tres posibles valores: 2017, 2018 o 2019.

Todas las variables mencionadas anteriormente se graficaron contra el número de viajes realizados, resultando en varias gráficas (Véase capítulo de Resultados) que fueron analizadas y ayudaron a tener información clave del sistema Bicipuma.

Variable dependiente N, análisis de respuesta del modelo

Como se ha indicado anteriormente, la variable dependiente N es el total de viajes realizados en un intervalo de 30 minutos de la estación *i* a la estación *j*, por ende, 182 enlaces fueron calculados y analizados individualmente.

Para este análisis se graficaron el total de solicitudes y llegadas durante el día por cada estación para entender los patrones de demanda diarios. Sin embargo, el análisis solo muestra el uso de bicicletas de una estación aislada. Es por esto que, a modo de ejemplo, las solicitudes de bicicletas de la estación 1 y sus interacciones con otras estaciones se graficaron para así poder observar cómo cada bicicleta solicitada en esta estación fue distribuida por los usuarios hacia otras estaciones de la red. Adicionalmente, se graficaron las llegadas a la estación 1 desde el resto de los nodos de la red para tener una perspectiva completa de las interacciones entre todas las estaciones Bicipuma.

Para ayudar a una mejor comprensión de lo recién descrito, pueden observarse en la *Figura 4.8* las solicitudes de la estación 9 y sus interacciones con el resto del sistema.

Análisis del modelo

Para un problema complejo con muchas variables por analizar, es importante comparar diferentes modelos de predicción. Por lo que a continuación se describen los modelos de predicción que se probaron en esta investigación.

Como primer modelo predictivo se probó el modelo de regresión múltiple. La regresión múltiple se construyó en el software R, tomando 80% de los datos totales para el proceso de modelado.

Algunas variables fueron removidas por ser estadísticamente no significativas, en otras palabras, el valor *p* era mayor al 5%. Este modelo resultó en la expresión siguiente *(Ecuación 1)*:

$$N = -222.8 + 0.01098(Precipitación\ total) + 0.01789(Temperatura\ promedio) - 0.003471(Humedad\ relativa) + 0.1101(Año) + 0.6279\ (Semestre) + 0.08142\ (Mes) - 0.02403(Lunes) + 0.04894\ (Martes) + 0.01635(Miércoles) + 0.02202(Jueves) + 0.01017(6{:}30) + 0.8811(7{:}00) + 0.2665(7{:}30) + 0.236(8{:}00) + 0.3257(8{:}30) + 0.7015(9{:}00) + 0.4397(9{:}30) + 0.5721\ (10{:}00) + 0.5774(10{:}30) + 1.019(11{:}00) + 0.8078(11{:}30) + 0.814(12{:}00) + 0.9319(12{:}30) + 1.63(13{:}00) + 1.284(13{:}30) + 1.262(14{:}00) + 1.214(14{:}30) + 1.447(15{:}00) + 0.9109(15{:}30) + 0.7244(16{:}00) - 0.4189(17{:}00) - 0.4054(17{:}30) \quad (1)$$

Puede observarse que el modelo no considera las variables de velocidad del viento, viernes, semana, y el intervalo 16:30. Se utilizó R-cuadrada determinar si el ajuste del modelo lineal era apropiado, obteniendo un resultado de 0.06024 donde se aprecia un ajuste insuficiente con las observaciones reales. Por este motivo, se utilizó el mismo método para el modelado del movimiento de las bicicletas de una estación específica a otra, mejorando el resultado a 0.3735. Sin embargo, este resultado aún muestra un limitado ajuste a los datos. Consecuentemente, se aplicaron los algoritmos Random Forests y XGBoost a estos datos, así como con la regresión múltiple, Random Forests y XGBoost fueron programados con software R.

Cabe mencionar que, R es un lenguaje de programación y un entorno de cómputo estadístico gráfico que proporciona una amplia variedad de técnicas gráficas y estadísticas. R puede expandirse a través de paquetes que cubren un gran rango de estadística moderna y otros temas (R Core Team, 2016).

Específicamente, se utilizó el paquete MLR3 para Random Forests y XGBoost, el cual define (Lang et al., 2019) como una herramienta que provee un marco genérico, extensible y orientado a objetos para clasificación, regresión, análisis de supervivencia y otras tareas de *Machine learning* para el lenguaje R. El proceso de modelado para Random Forests y XGBoost fueron muy similares.

A continuación, se presenta una breve explicación del proceso de modelado de esta investigación, específicamente con ambos algoritmos configurados para obtener un error mínimo, luego, mejores predicciones.

Se emplearon Random Forests y XGBoost para predecir el número de viajes estimados en cada enlace de la red cada 30 minutos. En otras palabras, el modelo calculó 182 resultados por cada intervalo de 30 minutos, es decir, 182 subconjuntos de datos fueron sujetos al proceso de modelado. Inicialmente, cada conjunto de datos se dividió en un 80% para datos de entrenamiento y 20% para datos de prueba. Primero, ambos modelos fueron probados con los hiperparámetros por defecto (definidos como Random Forests y XGBoost), después los datos de entrenamiento se procesaron para obtener los mejores hiperparámetros del modelo con error mínimo. Estos hiperparámetros se aplicaron a los 182 conjuntos de datos (definidos por el autor como Random Forest Tuned y XGBoost Tuned). Finalmente, los datos de entrenamiento se procesaron para obtener los mejores hiperparámetros de cada modelo (definidos por el autor como Random Forests Special Tuning y XGBoost Special Tuning).

Tipos de optimización de hiperparámetros:

- Random Forests, hiperparámetros por defecto (Num Trees 500, mtry 6)
- XGBoost, hiperparámetros por defecto (nrounds = 1, eta = 0.3, max_depth = 6)
- Random Forest Tuned (Num Trees 750, mtry 7)
  Mejores hiperparámetros en 182 resultados
- XGBoost Tuned
  Mejores hiperparámetros en 182 resultados
- Random Forests Special Tuning (Combinación única de hiperparámetros para 182 modelos)
- XGBoost Special Tuning (Combinación única de hiperparámetros para 182 modelos)

Los últimos dos tipos de optimización de hiperparámetros fue obtenido mediante un proceso iterativo, con un remuestreo de validación cruzada de *K-folds*, con 10 particiones por conjunto de datos, iterando los hiperparámetros dentro de un rango de 400-800 para el número de árboles, 4-8

para mtry en el caso de Random Forests y 0-0.5 para eta, 3-6 para max_depth y 100-160 para nrounds en XGBoost.

Los hiperparámetros modifican el proceso de aprendizaje de ambos algoritmos y con ello, la exactitud de la predicción. Todos los hiperparámetros que fueron modificados en el proceso de optimización se describen brevemente a continuación:

- Random Forests
    - Número de árboles (Ntree): Número de árboles en el modelo
    - Mtry: Número de variables muestreadas al azar como candidatas en cada división.

- XGBoost
    - Número de procesos de *boosting*: Número de procesos de *boosting* o árboles a construir
    - Eta: Este hiperparámetro determina el índice de aprendizaje, este corresponde a la reducción de variables asociadas a las características de cada ronda, es decir, define la "corrección" realizada en cada paso.
    - Max_depth: Se refiere al máximo número de nodos permitidos desde la raíz hasta la hoja más remota de un árbol. Árboles más complejos pueden modelar relaciones más sofisticadas al agregar más nodos, pero a mayor complejidad, las ramificaciones pierden relevancia aunada al ruido, provocando que el modelo se sobreajuste.

Figura 3.3 Parte del código en R, donde el paquete MLR3 es utilizado para entrenar y predecir el movimiento de las bicicletas para los 182 enlaces con Random Forests Special Tuning.

```
for (i in 1:length(y)) {

  testF <- read.csv(paste0("D:/Desktop/Tesis 2020 PC/bicipumas/NetSystemX/", y[i]  ,".csv"))
  testF <- testF[, -1]
  task_bikes <- TaskRegr$new(id = "bikes", backend = testF, target = "N")
  learner <- mlr_learners$get("regr.ranger")
  learner$param_set$values <- list(num.trees = Bnum.trees[i], mtry = Bmtry[i])

  indice <- task_bikes$nrow
  train_set <- 1:ceiling(indice*0.8)
  test_set <- setdiff(seq_len(task_bikes$nrow), train_set)

  learner$train(task_bikes, row_ids = train_set)
  prediction <- learner$predict(task_bikes, row_ids = test_set)

  prediction <- as.data.table(prediction)
  prediction$route <- y[i]
  TotalPredictions <- rbind(TotalPredictions, prediction)
}
```

Recuperado: Elaboración propia

Evaluación

Un paso clave en el proceso de ciencia de datos es el de evaluación del modelo. El objetivo es evaluar el desempeño del modelo o su fiabilidad para predecir resultados. En esta investigación se eligió un modelo popular de métrica de regresión para *Machine learning*, Error cuadrático medio o Root Mean Squared Error (RMSE). RMSE fue seleccionado para evaluar modelos por una simple razón, pues el resultado de RMSE se interpreta fácilmente puesto que sus resultados se expresan en las mismas unidades que la predicción de resultados. Dicho de otro modo, si el

resultado de RMSE de la predicción del modelo del sistema *bike-sharing* es 6.3, se puede interpretar que el modelo tiene un error de 6.3 bicicletas comparado con los datos reales (Datta, 2014).

Matemáticamente, el RMSE se calcula por medio de la siguiente fórmula *(Ecuación 2)*:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (2)$$

Donde *n* es el número de muestras de prueba, *y* las observaciones reales y $\hat{y}$ la proyección correspondiente (Wang and Kim, 2018).

Implementación y retroalimentación

Las etapas de implementación y retroalimentación se revisarán con los resultados de esta investigación comparando los resultados de las predicciones con los datos reales, mostrados gráfica y matemáticamente para así decidir si el modelo pudiera mejorar las decisiones del proceso de reabastecimiento del sistema *bike-sharing*, o no. Esto se explicará a detalle en la sección de resultados, además se discutirá en la sección de conclusiones, si la implementación del modelo es factible para la operación diaria de Bicipuma.

Capítulo 4

Resultados

Análisis de variables climátológicas

Como se mencionó anteriormente en la sección de modelado, todas las variables climatológicas fueron analizadas por medio de tres gráficas:

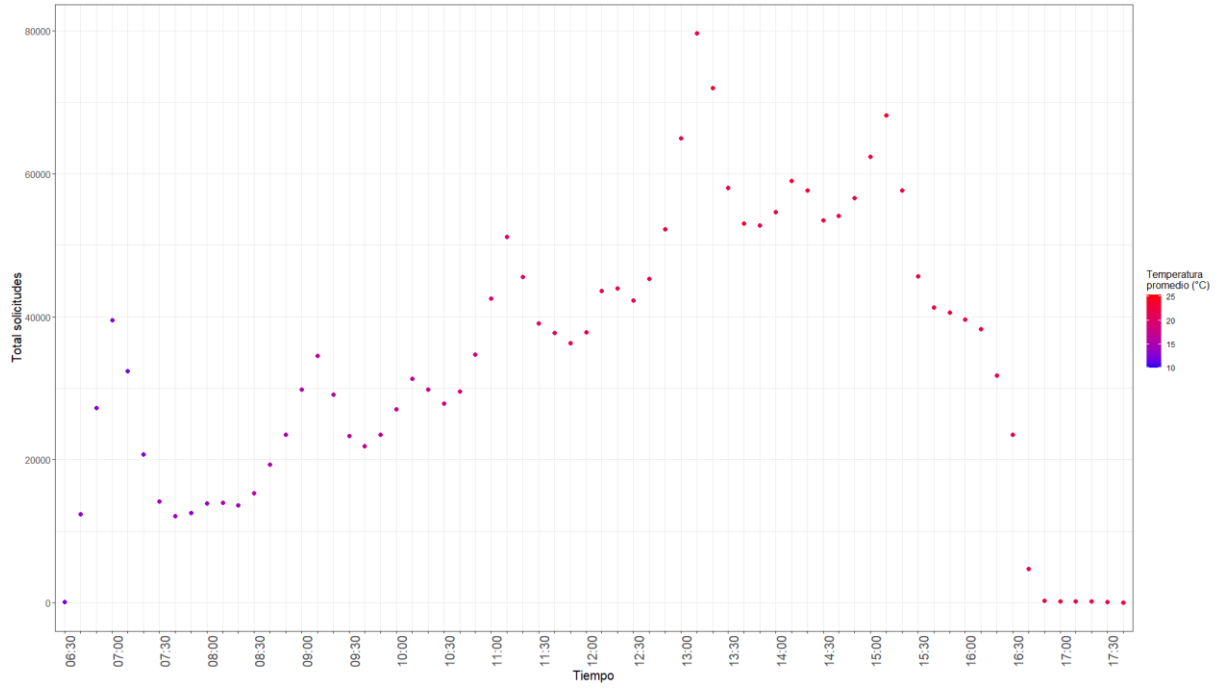
- Total de viajes realizados en un intervalo de 30 minutos[1] vs. una variable climatológica
- Total de viajes realizados en un intervalo de 30 minutos vs. una variable climatológica, agrupado por semana
- Total de viajes realizados en un intervalo de 30 minutos vs. tiempo de operación al día, y en color, los valores de la variable climatológica

Las tres gráficas fueron importantes para determinar si la variable afecta a la demanda del sistema o no. Como un ejemplo, la variable climatológica de temperatura fue graficada en relación con el total de viajes, como se puede ver en la *Figura 4.1*. Es importante mencionar que, la *Figura 4.1* muestra el total de viajes realizados en un intervalo de 10 minutos, con el propósito de tener una mejor comprensión de la relación entre la temperatura y la demanda del sistema.

Como se muestra en la *Figura 4.1*, hay una estacionalidad evidente en los movimientos diarios, presentando máximos de demanda cada dos horas, empezando desde 7 am a 3 pm. Asimismo, el total de los viajes en el sistema tiende a incrementar durante el día mientras la temperatura sube. Por esta razón, la variable temperatura fue considerada como un factor importante para el modelo de predicción.

[1] *Viajes totales realizados en un intervalo de 30 minutos, se refiere al conteo total de los datos disponibles desde 2017 a 2019, en intervalos de 30 minutos*. (Ejemplo: 6:30-7:00)

Figura 4.1. Solicitudes totales en intervalos de 10 minutos y su respectiva temperatura promedio registrada.



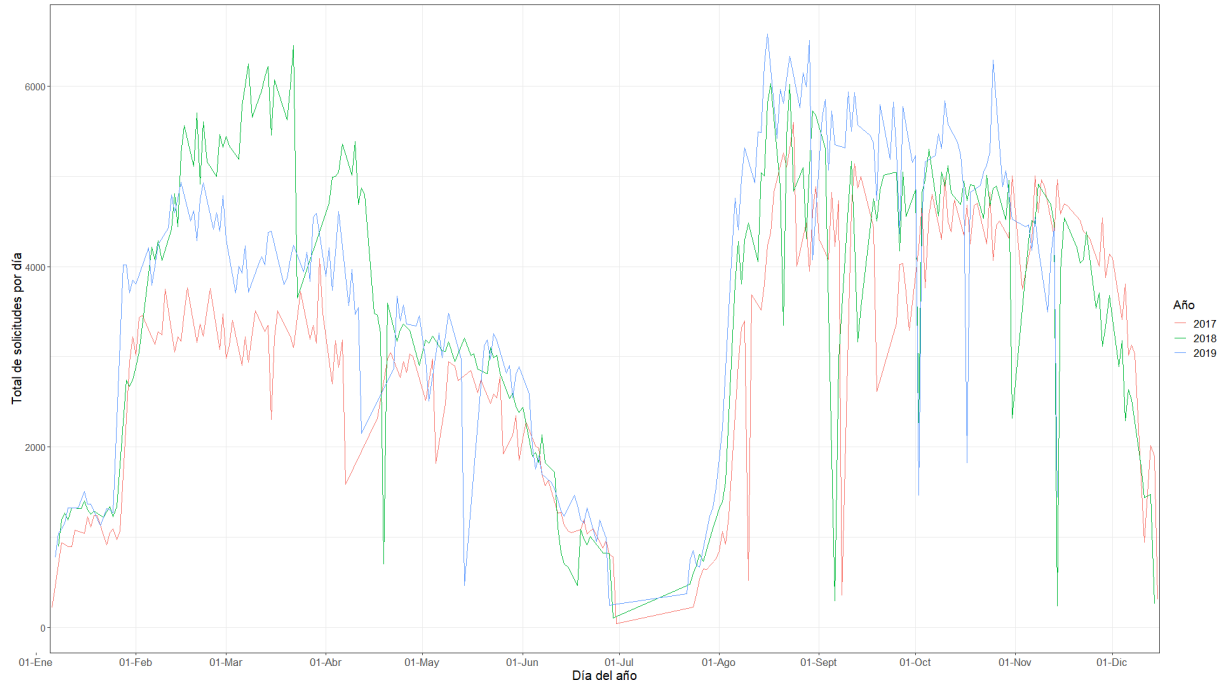
Recuperado: Elaboración propia utilizando el software R.

Al igual que el análisis de la variable temperatura, cada una de las variables climatológicas fueron analizadas y sometidas al mismo proceso y, al finalizar el proceso de análisis de todas las variables se concluyó que, la humedad relativa, velocidad de viento promedio y la precipitación total fueron consideradas en el modelo de predicción.

Análisis de la variable tiempo

Previamente dicho en esta investigación, el tiempo es una variable muy importante para el modelo de predicción porque, un simple análisis del primer gráfico creado para este trabajo fue descubierto un fuerte patrón de la demanda en relación con el tiempo. El gráfico antes mencionado puede ser visualizado a continuación, *(Figura 4.2)* donde un patrón muy similar se puede observar, comparando los tres años presentados (2017, 2018 y 2019).

Figura 4.2. Solicitudes totales registradas por día.



Recuperado: Elaboración propia utilizando el software R.

Otro hallazgo fue que, el volumen de los viajes realizados de un semestre a otro es muy distinto, por lo que es importante mencionar que, el semestre universitario de agosto-diciembre, una nueva generación de estudiantes ingresa a la universidad, así que la demanda de Bicipuma incrementa de manera significativa. Con este simple análisis, se concluyó que la variable Semestre es importante para el modelo de predicción. Además, se puede observar que hay un pequeño incremento en la demanda a través de los años, por lo tanto, la variable Año, de igual manera fue considerada.

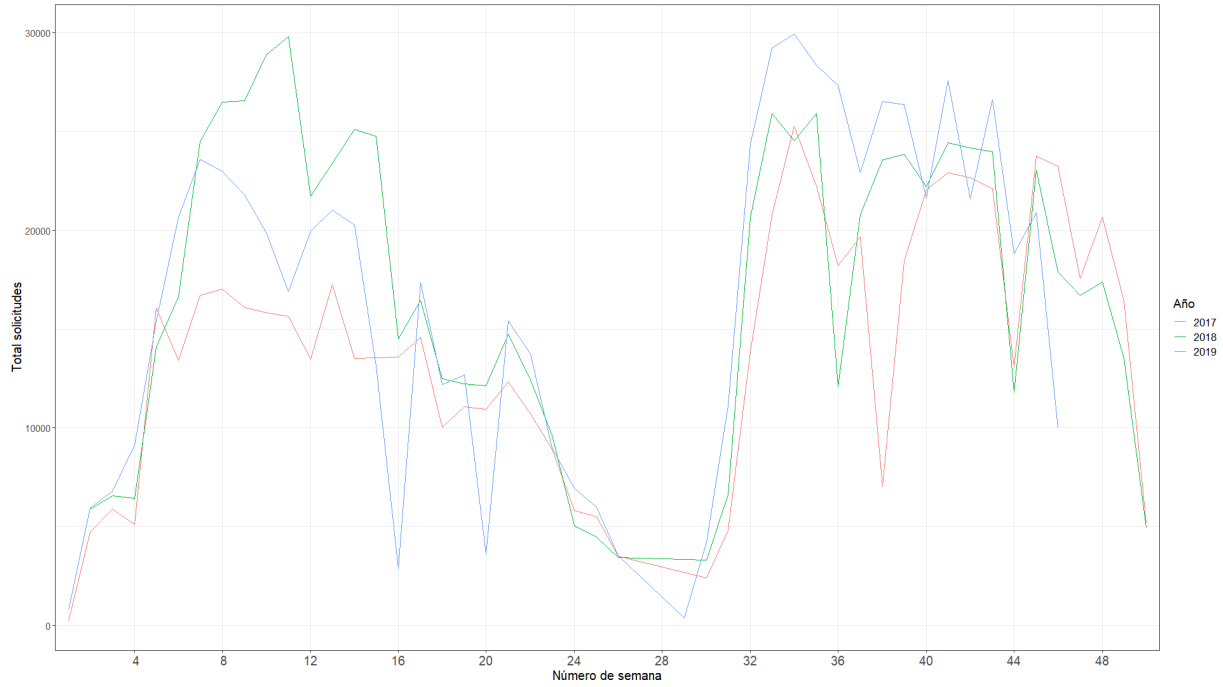
Siendo el lugar de estudio un campus universitario, es común observar que la afluencia de estudiantes varía durante el año, los meses y las semanas. El uso del servicio Bicipuma también puede variar durante meses con eventos especiales, festividades, ciertas actividades organizadas por los estudiantes, o simplemente porque el semestre ha comenzado o está finalizando. Todos estos factores pueden afectar a la demanda del sistema, es por eso que, la variable tiempo es muy importante para este estudio y fue dividida en algunas variables, incluyendo mes como se puede ver en la siguiente figura *(Figura 4.3)*.

Figura 4.3 Solicitudes totales registradas por mes.

Recuperado: Elaboración propia utilizando el software R.

Al igual que la variable Mes, el total de viajes fueron graficados en relación con el número de semanas del año (Comenzando desde 1 hasta 52) como se muestra en la *Figura 4.4*. Se puede observar una diferencia muy grande comparando una semana a otra. Uno de los factores importantes que puedan afectar la demanda semanal, podría ser los días de asueto, así como los mismos factores mencionados en el análisis de la variable Mes, por consiguiente, la variable semana también fue considerada para el modelo de predicción.

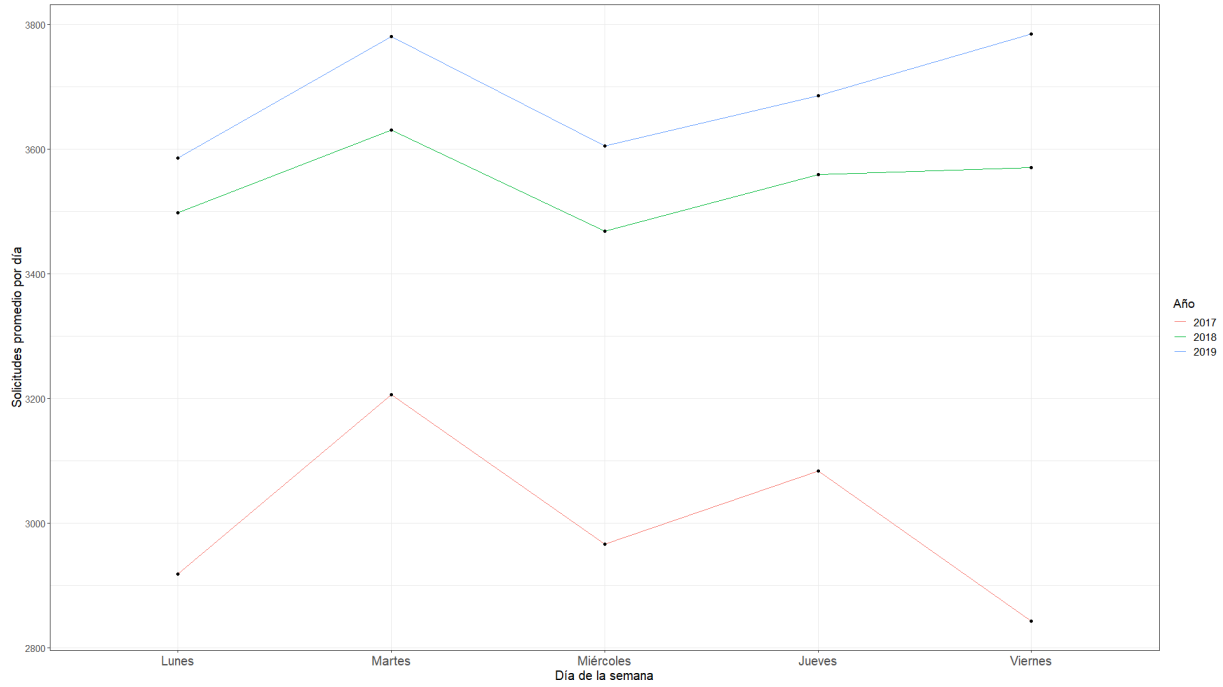
Figura 4.4. Solicitudes totales registradas por semana.



Recuperado: Elaboración propia utilizando el software R..

Por otra parte, los días de la semana también fueron analizados, con la hipótesis que se observarían diferentes niveles de demanda para cada día. Esta variable fue graficada en relación con los viajes totales del sistema, como se muestra en la *Figura 4.5*. Hay un patrón evidente entre los días de la semana, demostrando que los días martes, jueves y viernes son días con alta demanda para el sistema *bike-sharing* y una baja demanda para los días lunes y miércoles.

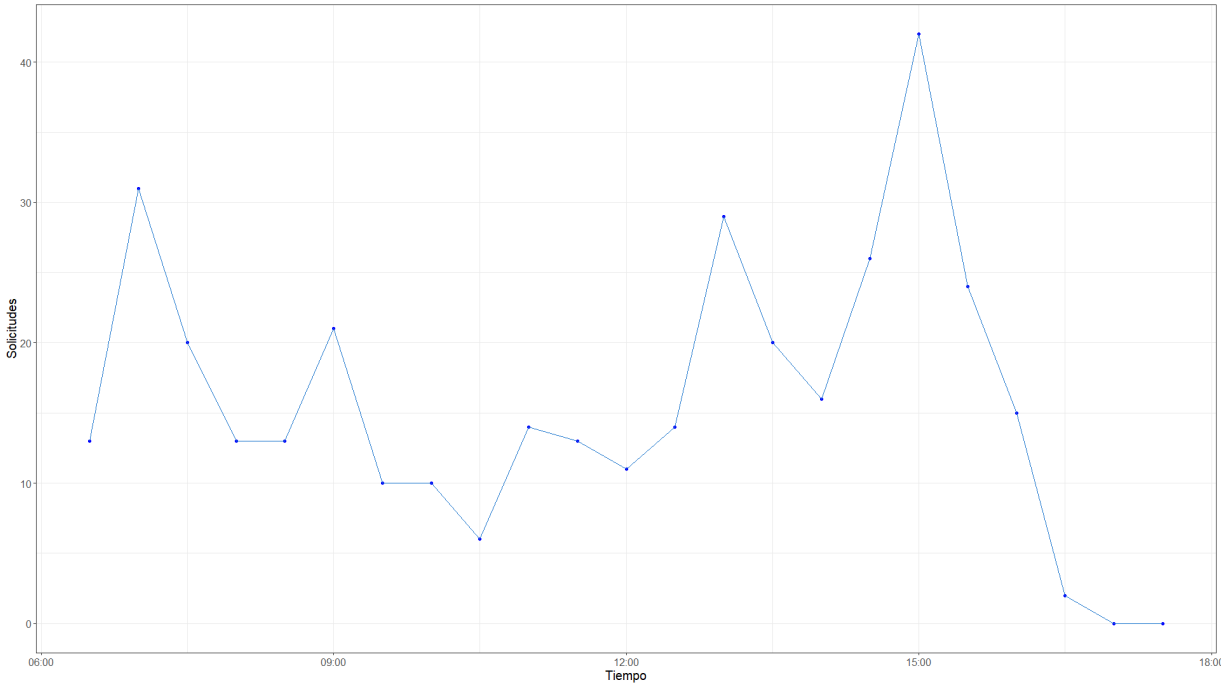
Figura 4.5. Promedio de solicitudes registradas por cada día de la semana.



Recuperado: Elaboración propia utilizando el software R.

Finalmente, en este último análisis presentado fue descubierto que la variable tiempo tiene una fuerte relación con los viajes realizados en el sistema en intervalos de 30 minutos, por lo tanto, se puede decir que el patrón de demanda se relaciona con las horas de clase tales como 7:00, 9:00, 11:00 y así sucesivamente. Así que se espera observar en la *Figura 4.6* algunos puntos altos de demanda en las horas mencionadas, ya que los estudiantes utilizan bicicletas del sistema para transportarse a su respectiva facultad. Por esta razón, dividir la variable de tiempo en periodos cortos, es fundamental para el modelo de predicción.

Figura 4.6. Solicitudes totales registradas el Lunes 30 de Septiembre del 2019, de la estación 1 a la 9.



Recuperado: Elaboración propia utilizando el software R.

Conforme a los descrito en esta sección, fue concluido que cada parte de la variable tiempo desde minutos hasta años (Minutos, días de la semana, semanas, meses, semestres y años) tienen una relación importante con el número de viajes realizados en el sistema, por lo cual todas estas variables fueron incluidas en el modelo de predicción.

Adicionalmente a todos los análisis previamente descritos, también fue considerada la importancia de cada variable para el modelo de prueba extraído de Random Forests para una comprensión sólida de las variables más importantes para el modelo. Todas las variables se muestran en la *Tabla 4.1*.

Tabla 4.1
Importancia de variables de Random Forests.

Variable	Importancia	Variable	Importancia
Temperatura_promedio	16.54%	Lunes	1.27%
Humedad_relativa	15.16%	int30_14_30	1.21%
Velocidad_viento_promedio	14.00%	int30_12_00	1.05%
Semana	9.61%	int30_15_30	1.04%
Mes	5.25%	Precipitación_total	0.77%
Año	3.82%	int30_16_00	0.77%
int30_13_30	3.25%	int30_11_30	0.72%
int30_13_00	2.95%	Semestre	0.72%
int30_17_00	2.36%	int30_10_30	0.69%
int30_17_30	2.25%	int30_11_00	0.65%
Viernes	2.10%	int30_10_00	0.49%
int30_12_30	1.73%	int30_09_30	0.45%
int30_14_00	1.70%	int30_06_30	0.45%
int30_15_00	1.51%	int30_08_30	0.41%
Jueves	1.49%	int30_09_00	0.36%
int30_16_30	1.45%	int30_07_30	0.32%
Martes	1.43%	int30_07_00	0.32%
Miércoles	1.40%	int30_08_00	0.31%

Recuperado: Elaboración propia.

Tabla 4.1 muestra el porcentaje de importancia de las 36 variables del modelo, es decir, si una variable tiene un mayor porcentaje significa que es una variable más importante para el modelo, de lo contrario, la variable es menos importante. Como resultado de este análisis de importancia, la temperatura, la humedad relativa y la velocidad del viento fueron las variables más importantes para el modelo, seguidas de las variables de tiempo como semana, mes y año. Desde una perspectiva analítica, el modelo puede alcanzar un 65% de precisión o, de modelado del sistema, solo con las seis variables mencionadas previamente.

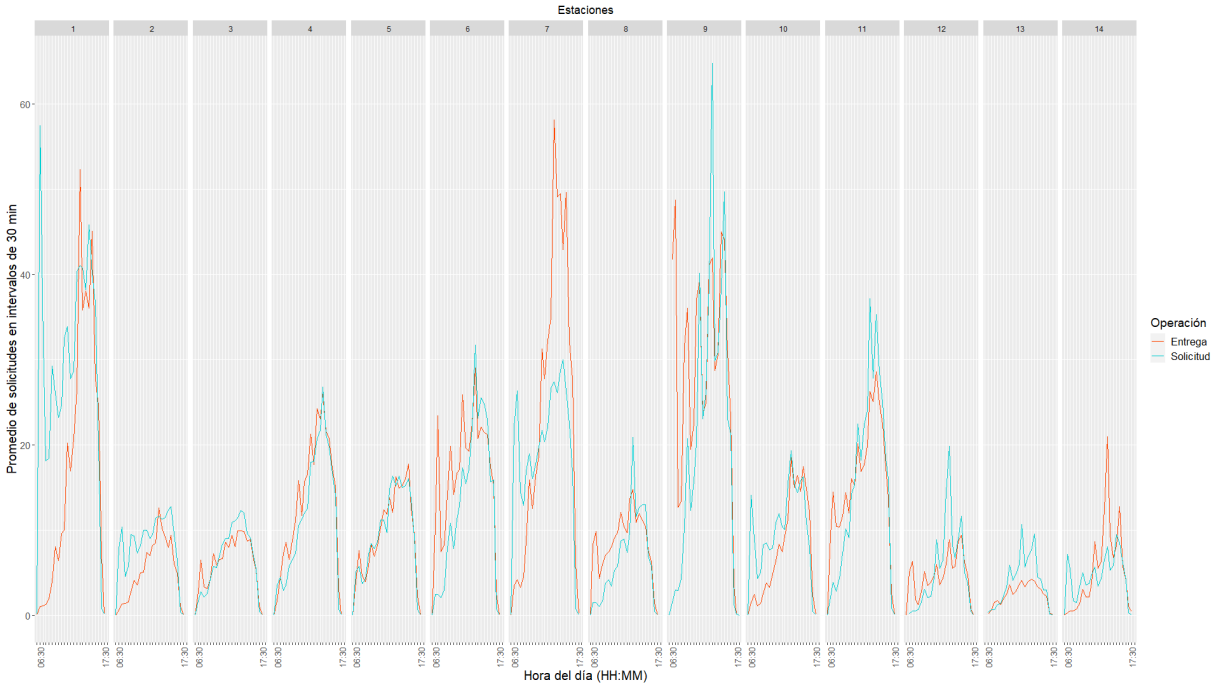
Para resumir, las variables que fueron analizadas y consideradas para el modelo de predicción están enlistadas a continuación:

- N
- Precipitación_total
- Temperatura_promedio
- Humedad_relativa
- Velocidad_viento_promedio
- Año
- Semestre
- Mes
- Semana
- Lunes
- Martes
- Miércoles
- Jueves
- Viernes
- int30_06_30
- int30_07_00
- int30_07_30
- int30_08_00
- int30_08_30
- int30_09_00
- int30_09_30
- int30_10_00
- int30_10_30
- int30_11_00
- int30_11_30
- int30_12_00
- int30_12_30
- int30_13_00
- int30_13_30
- int30_14_00
- int30_14_30
- int30_15_00
- int30_15_30
- int30_16_00
- int30_16_30
- int30_17_00
- int30_17_30

Variable dependiente N, análisis de respuesta del modelo

Figura 4.7 presenta el promedio de solicitudes (Azul) y entregas (Naranja) de bicicletas en intervalos de 30 minutos en cada estación de todos los datos disponibles. Esta gráfica muestra que el patrón de solicitudes y entregas tienen diferentes comportamientos en cada estación. Asimismo, el nivel de demanda es diferente entre las estaciones, algunas de ellas son más concurridas debido a su ubicación dentro del campus y su proximidad a los principales medios de transporte públicos como el metro o estaciones de autobuses.

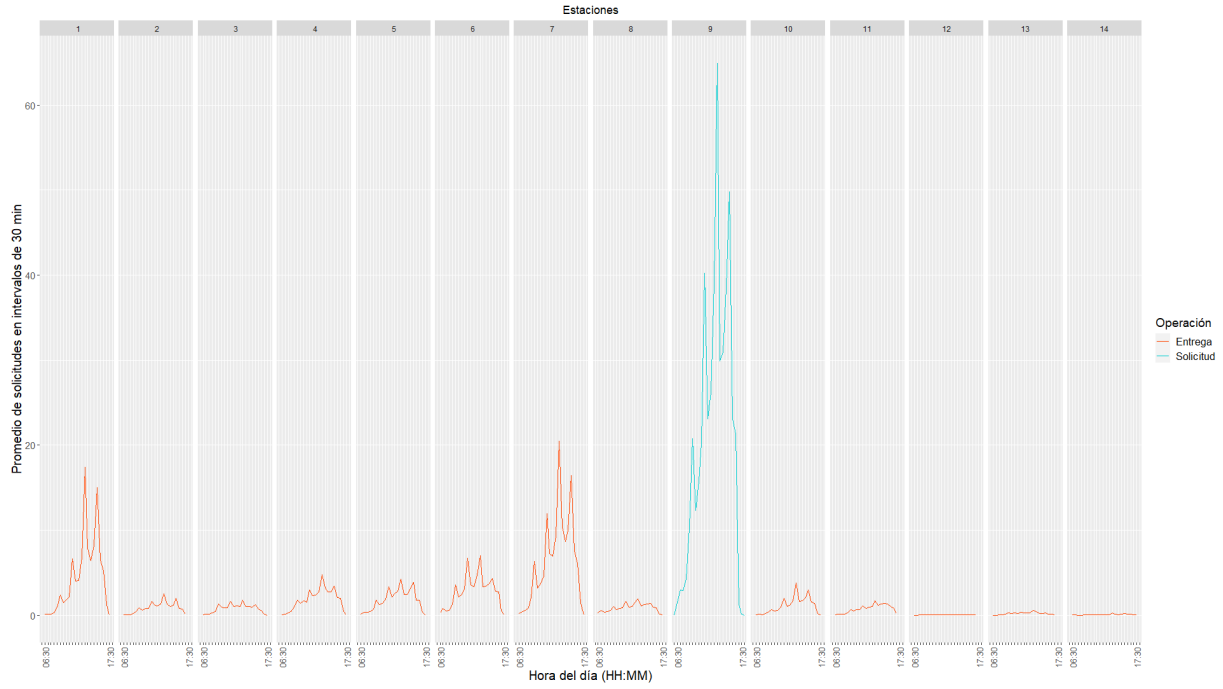
Figura 4.7. Solicitudes/Entregas registradas para un intervalo de 30 minutos para cada estación del sistema Bicipuma.



Recuperado: Elaboración propia utilizando el software R.

Para tener una mejor perspectiva de los movimientos entre cada una de las estaciones se realizó el análisis individual de una de las estaciones más concurridas en el sistema, la estación 9, este análisis consistió en entender su interacción con el resto de las estaciones. La *Figura 4.8* presenta los movimientos de bicicleta de la estación 9 a las estaciones 1,2,3, … 14, recalcando que hay un mayor flujo entre la estación 9 y las estaciones 1 y 7.

Figura 4.8. Promedio de solicitudes desde la estación 9 y las entregas de bicicletas registradas al resto de las estaciones del sistema, por cada intervalo de 30 minutos.



Recuperado: Elaboración propia utilizando el software R.

Por lo tanto, con los 2 análisis realizados se determinó que el número de viajes entre estaciones varía de acuerdo con la ubicación dentro del campus universitario y, que cada enlace de la red tiene su propio comportamiento, al igual, es importante considerar como una variable individual a cada una de las estaciones en el modelo de predicción.

Predicciones

Como fue descrito en la sección del análisis del modelo, seis modelos con diferentes hiperparámetros fueron comparados para tener una mejor predicción de la demanda del sistema. El promedio y la desviación estándar de los 182 resultados de RMSE obtenidos de los 182 enlaces de la red calculados de cada modelo, se exhiben en la *Tabla 4.2*.

Tabla 4.2

Promedio del RMSE y la desviación estándar del cálculo de los 182 enlaces de la red.

	RF Tuned	XGBoost Tuned	XGBoost	RF	RF Special Tuning	XGBoost Special Tuning
Average	1.180091	1.217222	1.579351	1.183184	1.179249	1.188588
Std dev	1.121092	1.136818	1.788515	1.134513	1.121387	1.122331

Recuperado: Elaboración propia.

En promedio, el menor RMSE presentado en la *Tabla 4.2* pertenece al modelo de Random Forests Special Tuning, además de que este modelo tuvo el mejor desempeño por el menor RMSE obtenido, en comparación a los otros cinco modelos, como se señala en la *Tabla 4.3*.

Tabla 4.3

Conteo total de los RMSE mínimos obtenidos por cada modelo de predicción.

Modelo de predicción	Conteo total de los RMSE mínimos
Random Forests	36
Random Forests Tuned	24
Random Forests Special Tuning	65
XGBoost	0
XGBoost Tuned	3
XGBoost Special Tuning	54

Recuperado: Elaboración propia.

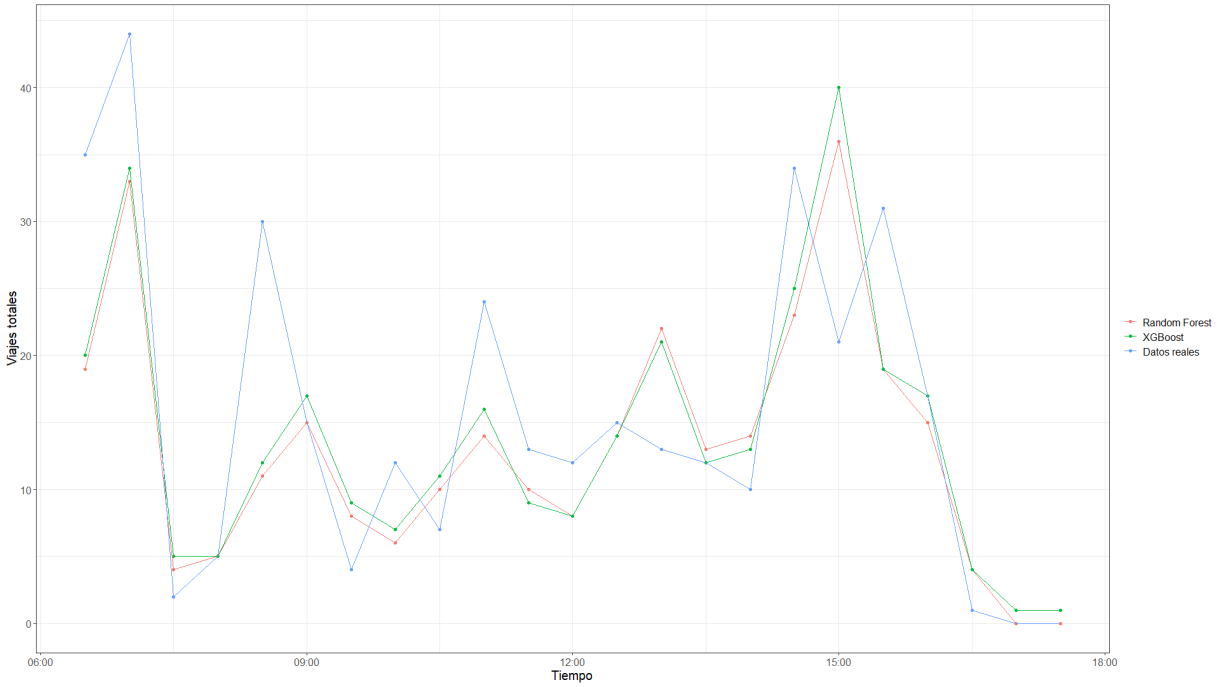
La *Tabla 4.3* se elaboró con el conteo del número de veces que un modelo tenía el menor RMSE en comparación a los otros modelos, es decir, si Random Forest Special tuning tuviera el menor

RMSE calculando el flujo de un enlace en específico, significaría que sumaría un uno a la cuenta de ese modelo. Y como resultado de la cuenta total de los RMSE, Random Forests Special Tuning y XGBoost Special Tuning fueron los dos modelos con mayores puntuaciones y por lo tanto, fueron seleccionados para esta investigación.

Con la selección de las variables más importantes, la optimización de hiperparámetros y el proceso de entrenamiento del 80% de los datos con Random Forests y XGBoost, los 182 modelos obtenidos fueron puestos a prueba en el 20% de los datos restantes, también conocidos como Test data.
La *Tabla 4.2*, muestra el promedio de los RMSE calculados de los 182 enlaces del sistema por ambos algoritmos de *Machine learning* usados para las predicciones finales. Estos valores de RMSE que se presentan en las últimas dos columnas de la *Tabla 4.2* están sesgados debido a la gran diferencia del nivel de demanda de cada estación o el flujo distinto en cada enlace del sistema. Para poder conseguir un enfoque sin sesgo y más comprensible, se analiza la predicción de los movimientos de la estación 1 a la 9, la cual es mostrada en la *Figura 4.9*. Cabe mencionar que este enlace es uno de los más concurridos en la red de Bicipuma.

Esta predicción se realizó bajo condiciones específicas para analizar la precisión del modelo, particularmente, la *Figura 4.9* presenta los viajes realizados desde la estación 1 a la 9, el lunes 7 de octubre del 2019, con una temperatura promedio durante el día de 18°C, humedad relativa de 50.94%, velocidad de viento promedio de 2.67 m/s y ninguna precipitación.

Figura 4.9. Predicción de los viajes por los algoritmos de Random Forests y XGBoost comparados con los viajes registrados (Datos reales).



Recuperado: Elaboración propia utilizando el software R.

Puede ser observado que hay algunas variaciones entre las predicciones de ambos algoritmos y los datos reales, pero las dos predicciones siguen el patrón de demanda de los movimientos diarios de los datos reales. El RMSE calculado por Random Forests en este día en específico es de 8.1560 y 8.0622 para XGBoost, por lo que significa que, en promedio, la predicción de ambos algoritmos tiene un error por ocho bicicletas en comparación a los datos reales. La desviación máxima de ambos algoritmos es de 19 bicicletas, es decir, en el intervalo de 8:00 am – 8:30 am la predicción del algoritmo Random Forests tiene un error de 19 bicicletas y, el mismo error se encuentra en el intervalo de 2:30 pm – 3:00 pm para XGBoost. Por otra parte, la desviación mínima es de cero o cercana a cero en algunos intervalos como se presenta en la *Figura 4.9*.

Debido a los resultados similares por parte de ambos algoritmos, se decidió utilizar los resultados de XGBoost para la predicción final de los viajes en bicicleta del sistema *bike-sharing*, ya que XGBoost es mucho más rápido al entrenar los datos y realizar predicciones en comparación a

Random Forests. La *Tabla 4.4* muestran los tiempos de entrenamiento y predicción, en segundos, de ambos algoritmos.

Tabla 4.4
Tiempo de predicción de los 182 modelos obtenidos por Random Forests y XGBoost.

Algoritmo de ML	Tiempo de entrenamiento (Seg)	Tiempo de predicción (Seg)
Random Forests	470.1	27.14
XGBoost	144.65	1.79

Recuperado: Elaboración propia.

Conclusión y recomendaciones para trabajos posteriores

Actualmente, la generación y recolección de datos es más grande que nunca a causa de los avances tecnológicos, por lo que cada proceso, actividad, y cada operación de un medio de transporte puede ser digitalizada, como consecuencia de ello, hay un gran volumen de datos obtenidos de cualquier sistema y estos, pueden ser utilizados para generar análisis, y así mejorar el desempeño de un sistema, reduciendo costos, optimizando el uso de los recursos, etcétera. A esto se le llama Industria 4.0, donde la automatización, la conectividad de los datos y herramientas como *Machine learning* son utilizados para operar dentro de la industria, usando dispositivos, máquinas y sistemas inteligentes que continuamente monitorean la producción y recolectan datos, y estos, son analizados para generar tableros en tiempo real y así tomar mejores decisiones (Ashmore, 2020)

La recolección de datos por un sistema *bike-sharing* es común en la actualidad, por lo tanto, hay muchas investigaciones acerca del análisis de estos datos ya que, hay un problema similar entre todos los sistemas *bike-sharing*, la escasez y los excedentes en algunas estaciones, lo que puede significar que los sistemas no están satisfaciendo la demanda, por ende, estos sistemas pueden ser mejorados.

En esta investigación fue descrito paso a paso el proceso de ciencia de datos para la caracterización y la predicción de la demanda de un sistema *bike-sharing* universitario llamado Bicipuma, empleando dos herramientas de *Machine learning* (Random Forests y XGBoost), y siguiendo una metodología de 10 pasos propuesta por (Rollins, 2015). Durante este proceso fueron analizadas y determinadas las variables climatológicas y otros factores que afectan a la demanda del sistema. Cada una de las variables fue analizada individualmente y se llegó a la conclusión que, dividir la variable de tiempo en minutos, días de la semana, semana, mes, semestre y año, fue crítico para el desarrollo del modelo. Al igual, las variables climatológicas como temperatura, humedad relativa, velocidad de viento promedio y precipitación total fueron consideradas para el modelo.

Un gran número de enfoques y perspectivas sobre el análisis de los datos generados por un sistema *bike-sharing* y sus respectivas predicciones fueron revisadas, como se puede observar en la sección de investigaciones relacionadas. Así que es importante mencionar que, al contrario de estos

enfoques revisados, esta investigación se concentra en predecir los viajes realizados de una estación a otra, por lo que, con 14 estaciones y 182 enlaces, se realizaron 182 modelos y la base de datos general fue divida en 182 secciones, con el único objetivo de facilitar el análisis para esta y futuras investigaciones.

De una manera más concreta, Random Forests y XGBoost fueron empleados para la predicción del número de viajes en cada enlace de la red Bicipuma en intervalos de 30 minutos, considerando el 80% del total de los datos recolectados por el sistema Bicipuma para el entrenamiento de los 2 algoritmos antes mencionados y, el 20% de los datos para poner a prueba el modelo y realizar predicciones de la demanda. Random Forests y XGBoost realizaron predicciones de la demanda con un RMSE similar, sin embargo, debido a una mejor eficiencia computacional, XGBoost fue seleccionado, con un RMSE de 1.18 y 6.8 para uno de los enlaces de la red más concurridos.
Sin embargo, como se muestra en la *Figura 4.9*, el RMSE de la predicción de los movimientos o viajes diarios de un enlace en específico es de 8.0622, con una variación mínima de 0 y una máxima de 19, resultando en una predicción aceptable para un sistema *bike-sharing* con 14 estaciones y más de 4,000 viajes diarios.

Como un sistema *bike-sharing* universitario, sería recomendable registrar y documentar eventos especiales en el campus, días de asueto, huelgas estudiantiles por facultad y periodos donde Bicipuma está fuera de servicio por condiciones climáticas adversas o por otras razones, ya que todo lo anterior afecta a la demanda de Bicipuma y es fundamental considerar todos estos tipos de variables para un modelado más preciso del sistema.

Esta investigación presenta una caracterización formal de la demanda de Bicipuma la cuál obtiene puntos clave del comportamiento de la demanda, así como la relación de las variables o factores que impactan a la demanda de Bicipuma, estos dos resultados son fundamentales para tomar mejores decisiones sobre las operaciones de Bicipuma. Adicionalmente, esta investigación es el primer paso para construir un sistema de abastecimiento automático y transformar a Bicipuma o cualquier otro sistema *bike-sharing* en un sistema de transporte inteligente (ITS).

Referencias

Anónimo. (Septiembre, 2020). Random Forest Explained: Random Forest explained simply*: An easy Introduction to Training, Classification, and Regression.* Towards data science. Recuperado de: https://towardsdatascience.com/random-forest-explained-7eae084f3ebe

Ashqar, H. I., Elhenawy, M., Almannaa, M. H., Ghanem, A., Rakha H. A. & House, L. (2017). Modeling bike availability in a bike-sharing system using machine learning. 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), 374-378. http://dx.doi.org/10.1109/MTITS.2017.8005700

Ashmore, H. (2020). Industry 4.0 and the Impacts of Machine Learning on the Manufacturing Industry. AiThority. Recuperado de: https://aithority.com/machine-learning/industry-4-0-and-the-impacts-of-machine-learning-on-the-manufacturing-industry/

Brownlee, J. (Marzo, 2021). XGBoost for regression. Machine learning mastery. Recuperado de: https://machinelearningmastery.com/xgboost-for-regression/

Brownlee, J. (Julio, 2020). Train-Test Split for Evaluating Machine Learning Algorithms. Machine learning mastery. Recuperado de: https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/

Brownlee, J. (Agosto, 2016). A Gentle Introduction to XGBoost for Applied Machine Learning. Machine learning mastery. Recuperado de: https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

Datta, A. K. (2014) Predicting bike-share usage patterns with machine learning, Master's Thesis, Department of informatics, University of Oslo.

DeMaio, P. (2009). Bike-sharing: History, Impacts, Models of Provision, and Future. *Journal of public transportation.* 12, 41-56. http://doi.org/10.5038/2375-0901.12.4.3

Embarq network. (2021). From Amsterdam to Beijing: The Global Evolution of Bike Share. Industry dive. Recuperado de: https://www.smartcitiesdive.com/ex/sustainablecitiescollective/amsterdam-beijing-global-evolution-bike-share/1100421/

Ghatak, A. (2017). Machine learning with R. Springer nature. 2017. https://doi.org/10.1007/978-981-10-6808-9

Hernandez, G. (2013) Propuesta de plan maestro de infraestructura ciclista para el campus de ciudad universitaria. División de ingenierías civil y geomática. Facultad de ingeniería, UNAM.

Hurwitz, J. & Kirsch, D. (2018). *Machine learning for dummies*. (IBM Limited Edition). New Jersey: John Wiley & Sons, Inc. ISBN: 978-1-119-45494-6

IBM cloud education. (Deciembre, 2020). Random Forest. IBM. Recuperado de: https://www.ibm.com/cloud/learn/random-forest

Institute for transportation & development policy (ITDP). (2018) *The bikeshare planning guide*. (2018 Edition). ITDP.

James, G., Witten, D., Hastie, T. & Tibshirani, R. (2013) *An introduction to statistical learning* with Applications in R. (1st Ed.) New York: Springer. http://dx.doi.org/10.1007/978-1-4614-7138-7

Kaggle. (2021). XGBoost. Kaggle. Recuperado de: https://www.kaggle.com/dansbecker/xgboost

Lang, M., Binder, M., Richter, J., Schratz, P., Pfisterer, F., Coors, S., Au, Q., Casalicchio, G., Kotthoff, L. & Bischl, B. (2019). *Journal of Open Source Software*. 4 (44), 1903. https://doi.org/10.21105/joss.01903

Malik, S., Harode, R. & Kunwar, A. S. (2020) XGBoost: A deep dive into boosting. *Simon Fraser University*. https://doi.org/10.13140/RG.2.2.15243.64803

Moncayo–Martínez L.A. (2020) Analysing Data Set of the Bike–Sharing System Demand with R Scripts: Mexico City Case. In: Bi Y., Bhatia R., Kapoor S. (eds) Intelligent Systems and Applications. IntelliSys 2019. Advances in Intelligent Systems and Computing. 1038, (90-105). https://doi.org/10.1007/978-3-030-29513-4_7

Mujtaba, H. (Septiembre, 2020). What is Cross Validation in Machine learning? Types of Cross Validation. Great learning. Recuperado de: https://www.mygreatlearning.com/blog/cross-validation/

O´Brien, O. (2014). Bicycle sharing systems - Global trends in size, *Centre for Advanced Spatial Analysis (CASA)*, University College London, May (196). ISSN: 1467-1298

Olvera, V., García, C., Pérez, A., Chagala, Y., Wellens, A. & Segura, E. (2018) Demand analysis for the BICIPUMA bike-sharing system in UNAM-MEXICO. International Congress on Logistics & Supply Chain (CiLOG).

R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Recuperado de: http://www.R-project.org/

Red Universitaria de Observatorios Atmosféricos de la Universidad Nacional Autónoma de México. (2021). Base de datos meteorológicos pública. Recuperado de: https://www.ruoa.unam.mx/index.php?page=estaciones&id=1

Rollins, J. B. (2015) Foundational methodology for data science, IBM Analytics, IBM.

Schuijbroek, J., Hampshire, R.C. & van Hoeve, W.-J. (2017). Inventory rebalancing and vehicle routing in bike sharing systems. *European Journal of Operational Research*, 257 (3), 992-1004. https://doi.org/10.1016/j.ejor.2016.08.029.

Shaheen, S. A., Guzman S. & Zhang, H. (2010). Bikesharing in Europe, the Americas, and Asia: Past, Present, and Future. *Transportation Research Record*. 2143(1), 159-167. http://dx.doi.org/10.3141/2143-20

Sickles, R., & Zelenyuk, V. (2019). Measurement of Productivity and Efficiency: Theory and Practice. *Cambridge: Cambridge University Press.* Marzo 2019. https://doi.org/10.1017/9781139565981

Travesía UNAM. (2020). Recuperado de: https://travesiaunam.com/que-es-bicipuma/

The observer. (2018). Renueva y moderniza la UNAM el sistema Bicipuma. The observer: *periodismo y verificador del discurso político.* Recuperado: https://www.theobserver.mx/2019/02/11/renueva-y-moderniza-la-unam-el-sistema-bicipuma/

UNAM. (2018). *Dirección general de comunicación social.* Recuperado de: https://www.dgcs.unam.mx/boletin/bdboletin/2018_251.html

UNAM. (2019). *Dirección general de comunicación social*. Recuperado de: https://www.dgcs.unam.mx/boletin/bdboletin/2019_192.html

UNAM. (2017). *Dirección general de servicios generales y movilidad: Bicipuma*. Recuperado de: https://www.dgsgm.unam.mx/bicipuma

Verikas, A., Vaiciukynas, E., Gelzinis, A., Parker, J. & Olsson, C. (2016). Electromyographic Patterns during Golf Swing: Activation Sequence Profiling and Prediction of Shot Effectiveness. Sensors. 16 (592). https://doi.org/10.3390/s16040592

Wang, B. & Kim, I. (2018) Short-term prediction for bike-sharing service using machine learning. *Transportation Research Procedia*. 34 (2018) 171–178. https://doi.org/10.1016/j.trpro.2018.11.029

Wang, W. (2016). Forecasting Bike Rental Demand Using New York Citi Bike Data. *School of computing, Technological University Dublin*. Enero 2016.

Wickham, H. (2014). Tidy data. *The Journal of Statistical Software*. 59 (10), 2014. https://doi.org/10.18637/jss.v059.i10

Wood, T. (2021). What is a Random Forest?. Deep Ai. Recuperado de: https://deepai.org/machine-learning-glossary-and-terms/random-forest

Yang, Y., Heppenstall, A, Turner, A & Comber, A. Using graph structural information about flows to enhance short-term demand prediction in bike-sharing systems. *Computers, Environment and Urban Systems*. 83 (2020). https://doi.org/10.1016/j.compenvurbsys.2020.101521

Yiu, T. (Junio, 2019). Understanding Random Forest: How the Algorithm Works and Why it Is So Effective. Towards data science. Recuperado de: https://towardsdatascience.com/understanding-random-forest-58381e0602d2